Signals and Systems

with MATLAB ® Computing and Simulink ® Modeling

Fourth Edition

Steven T. Karris

$$X[m] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{mn}{N}}$$

Includes step-by-step procedures for designing analog and digital filters

Orchard Publications
www.orchardpublications.com

Signals and Systems

with MATLAB ® Computing and Simulink ® Modeling

Fourth Edition

Students and working professionals will find Signals and Systems with MATLAB ® Computing and Simulink ® Modeling, Fourth Edition, to be a concise and easy-to-learn text. It provides complete, clear, and detailed explanations of the principal analog and digital signal processing concepts and analog and digital filter design illustrated with numerous practical examples.

This text includes the following chapters and appendices:

• Elementary Signals • The Laplace Transformation • The Inverse Laplace Transformation • Circuit Analysis with Laplace Transforms • State Variables and State Equations • The Impulse Response and Convolution • Fourier Series • The Fourier Transform • Discrete Time Systems and the Z Transform • The DFT and The FFT Algorithm • Analog and Digital Filters • Introduction to MATLAB ® • Introduction to Simulink ® • Review of Complex Numbers • Review of Matrices and Determinants • Window Functions

Each chapter contains numerous practical applications supplemented with detailed instructions for using MATLAB and Simulink to obtain accurate and quick solutions.

Steven T. Karris is the founder and president of Orchard Publications, has undergraduate and graduate degrees in electrical engineering, and is a registered professional engineer in California and Florida. He has more than 35 years of professional engineering experience and more than 30 years of teaching experience as an adjunct professor, most recently at UC Berkeley, California. His area of interest is in The MathWorks, Inc. ™ products and the publication of MATLAB® and Simulink® based texts.

Orchard Publications

Visit us on the Internet
www.orchardpublications.comor email us:
info@orchardpublications.com

ISBN-10: 1-934404-12-8

ISBN-13: 978-1-934404-12-6

$70.00 U.S.A.

Signals and Systems

with MATLAB® Computing and Simulink® Modeling

Fourth Edition

Steven T. Karris

Orchard Publications
www.orchardpublications.com

Signals and Systems with MATLAB® Computing and Simulink Modeling®, Fourth Edition

Direct all inquiries to Orchard Publications, info@orchardpublications.com

Library of Congress Cataloging-in-Publication Data

Catalog record is available from the Library of Congress

Library of Congress Control Number: 2008927083

ISBN–13: 978–1–934404–12–6

ISBN–10: 1–934404–12–8

Copyright TX 5–471–562

Preface

This text contains a comprehensive discussion on continuous and discrete time signals and systems with many MATLAB® and several Simulink® examples. It is written for junior and senior electrical and computer engineering students, and for self–study by working professionals. The prerequisites are a basic course in differential and integral calculus, and basic electric circuit theory.

This book can be used in a two–quarter, or one semester course. This author has taught the subject material for many years and was able to cover all material in 16 weeks, with 2½ lecture hours per week.

To get the most out of this text, it is highly recommended that Appendix A is thoroughly reviewed. This appendix serves as an introduction to MATLAB, and is intended for those who are not familiar with it. The Student Edition of MATLAB is an inexpensive, and yet a very powerful software package; it can be found in many college bookstores, or can be obtained directly from

The MathWorks™ Inc., 3 Apple Hill Drive, Natick, MA 01760–2098
Phone: 508 647–7000, Fax: 508 647–7001
http://www.mathworks.com
e–mail: info@mathworks.com

The elementary signals are reviewed in Chapter 1, and several examples are given. The purpose of this chapter is to enable the reader to express any waveform in terms of the unit step function, and subsequently the derivation of the Laplace transform of it. Chapters 2 through 4 are devoted to Laplace transformation and circuit analysis using this transform. Chapter 5 is an introduction to state–space and contains many illustrative examples. Chapter 6 discusses the impulse response. Chapters 7 and 8 are devoted to Fourier series and transform respectively. Chapter 9 introduces discrete–time signals and the $\mathcal{Z}$ transform. Considerable time was spent on Chapter 10 to present the Discrete Fourier transform and FFT with the simplest possible explanations. Chapter 11 contains a thorough discussion to analog and digital filters analysis and design procedures. As mentioned above, Appendix A is an introduction to MATLAB. Appendix B is an introduction to Simulink, Appendix C contains a review of complex numbers, and Appendix D is an introduction to matrix theory.

New to the Second Edition

This is an extensive revision of the first edition. The most notable change is the inclusion of the solutions to all exercises at the end of each chapter. It is in response to many readers who expressed a desire to obtain the solutions in order to check their solutions to those of the author and thereby enhancing their knowledge. Another reason is that this text is written also for self–

study by practicing engineers who need a review before taking more advanced courses such as digital image processing.

Another major change is the addition of a rather comprehensive summary at the end of each chapter. Hopefully, this will be a valuable aid to instructors for preparation of view foils for presenting the material to their class.

New to the Third Edition

The most notable change is the inclusion of Simulink modeling examples. The pages where they appear can be found in the Table of Contents section of this text. Another change is the improvement of the plots generated by the latest revisions of the MATLAB® Student Version, Release 14.

The author wishes to express his gratitude to the staff of The MathWorks™, the developers of MATLAB® and Simulink®, especially to Ms. Courtney Esposito, for the encouragement and unlimited support they have provided me with during the production of this text.

Our heartfelt thanks also to Ms. Sally Wright, P.E., of Renewable Energy Research Laboratory University of Massachusetts, Amherst, for bringing some errors on the previous editions to our attention.

New to the Fourth Edition

The most notable change is the inclusion of Appendix E on window functions. The plots were generated generated with the latest revisions of the MATLAB® R2008a edition. Also, two end-of- chapter exercises were added in Chapter 10 to illustrate the use of the fft and ifft MATLAB functions

The author wishes to express his gratitude to the staff of The MathWorks™, the developers of MATLAB® and Simulink®, especially to The MathWorks™ Book Program Team, for the encouragement and unlimited support they have provided me with during the production of this and all other texts by this publisher.

Orchard Publications
www.orchardpublications.com
info@orchardpublications.com

Table of Contents

Simulink Modeling
Page 7–31

MATLAB Computing

Simulink Modeling

Chapter 1

Elementary Signals

This chapter begins with a discussion of elementary signals that may be applied to electric networks. The unit step, unit ramp, and delta functions are then introduced. The sampling and sifting properties of the delta function are defined and derived. Several examples for expressing a variety of waveforms in terms of these elementary signals are provided. Throughout this text, a left justified horizontal bar will denote the beginning of an example, and a right justified horizontal bar will denote the end of the example. These bars will not be shown whenever an example begins at the top of a page or at the bottom of a page. Also, when one example follows immediately after a previous example, the right justified bar will be omitted.

1.1 Signals Described in Math Form

Consider the network of Figure 1.1 where the switch is closed at time $t = 0$.

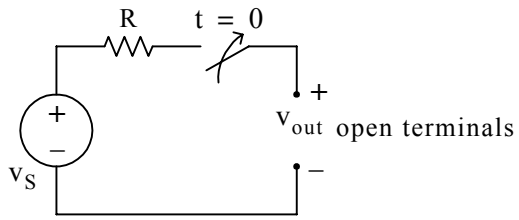


Figure 1.1. A switched network with open terminals

We wish to describe v_{out} in a math form for the time interval $-\infty < t < +\infty$. To do this, it is convenient to divide the time interval into two parts, $-\infty < t < 0$, and $0 < t < \infty$.

For the time interval $-\infty < t < 0$, the switch is open and therefore, the output voltage v_{out} is zero. In other words,

$$v_{out} = 0 \quad \text{for} \quad -\infty < t < 0 \tag{1.1}$$

For the time interval $0 < t < \infty$, the switch is closed. Then, the input voltage v_S appears at the output, i.e.,

$$v_{out} = v_S \quad \text{for} \quad 0 < t < \infty \tag{1.2}$$

Combining (1.1) and (1.2) into a single relationship, we obtain

$$v_{out} = \begin{cases} 0 & -\infty < t < 0 \\ v_S & 0 < t < \infty \end{cases} \tag{1.3}$$

We can express (1.3) by the waveform shown in Figure 1.2.

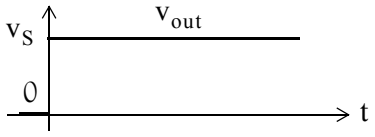


Figure 1.2. Waveform for v_{out} as defined in relation (1.3)

The waveform of Figure 1.2 is an example of a discontinuous function. A function is said to be *discontinuous* if it exhibits points of discontinuity, that is, the function jumps from one value to another without taking on any intermediate values.

1.2 The Unit Step Function $u_0(t)$

A well known discontinuous function is the *unit step function* $u_0(t)$ * which is defined as

$$u_0(t) = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \end{cases} \tag{1.4}$$

It is also represented by the waveform of Figure 1.3.

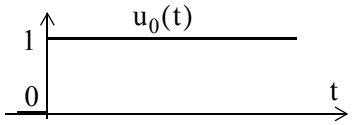


Figure 1.3. Waveform for $u_0(t)$

In the waveform of Figure 1.3, the unit step function $u_0(t)$ changes abruptly from 0 to 1 at $t = 0$. But if it changes at $t = t_0$ instead, it is denoted as $u_0(t - t_0)$. In this case, its waveform and definition are as shown in Figure 1.4 and relation (1.5) respectively.

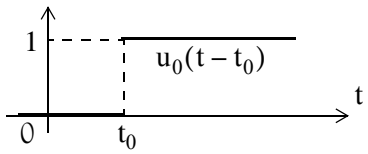


Figure 1.4. Waveform for $u_0(t - t_0)$

* *In some books, the unit step function is denoted as* $u(t)$, *that is, without the subscript 0. In this text, however, we will reserve the* $u(t)$ *designation for any input when we will discuss state variables in Chapter 5.*

$$u_0(t-t_0) = \begin{cases} 0 & t < t_0 \\ 1 & t > t_0 \end{cases} \tag{1.5}$$

If the unit step function changes abruptly from 0 to 1 at $t = -t_0$, it is denoted as $u_0(t+t_0)$. In this case, its waveform and definition are as shown in Figure 1.5 and relation (1.6) respectively.

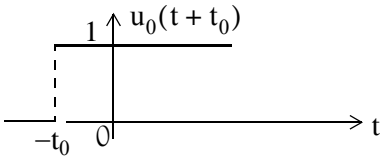


Figure 1.5. Waveform for $u_0(t+t_0)$

$$u_0(t+t_0) = \begin{cases} 0 & t < -t_0 \\ 1 & t > -t_0 \end{cases} \tag{1.6}$$

Example 1.1

Consider the network of Figure 1.6, where the switch is closed at time $t = T$.

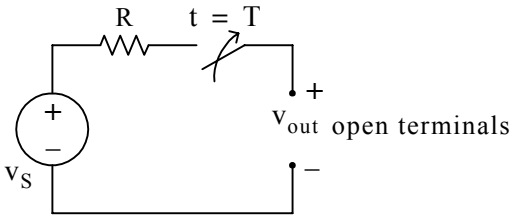


Figure 1.6. Network for Example 1.1

Express the output voltage v_{out} as a function of the unit step function, and sketch the appropriate waveform.

Solution:

For this example, the output voltage $v_{out} = 0$ for $t < T$, and $v_{out} = v_S$ for $t > T$. Therefore,

$$v_{out} = v_S u_0(t-T) \tag{1.7}$$

and the waveform is shown in Figure 1.7.

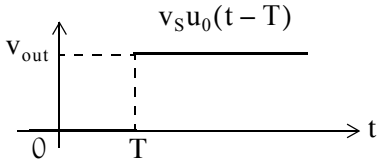


Figure 1.7. Waveform for Example 1.1

Other forms of the unit step function are shown in Figure 1.8.

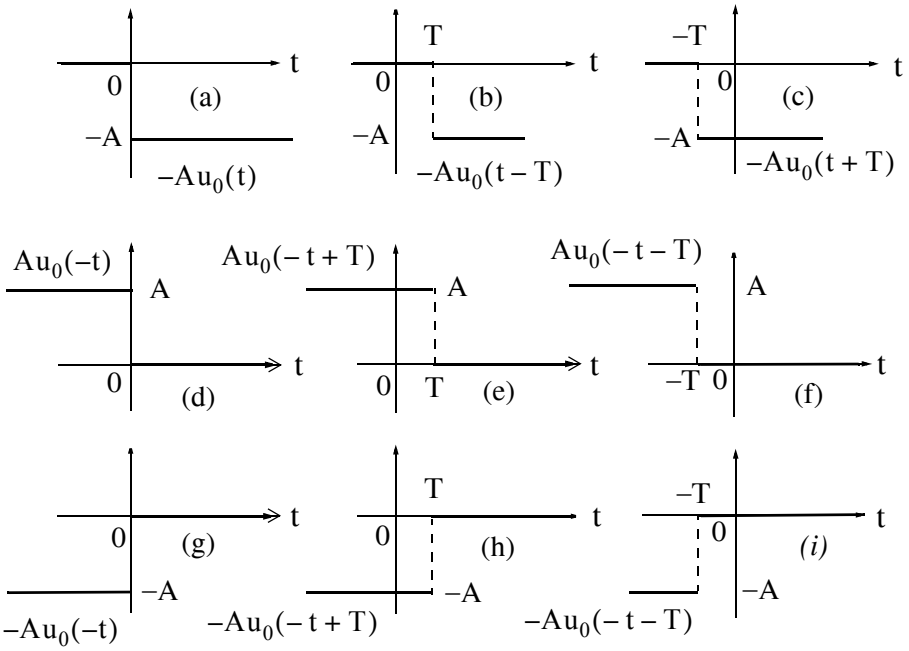


Figure 1.8. Other forms of the unit step function

Unit step functions can be used to represent other time–varying functions such as the rectangular pulse shown in Figure 1.9.

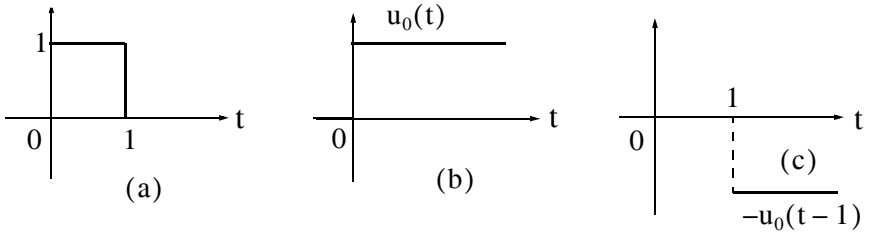


Figure 1.9. A rectangular pulse expressed as the sum of two unit step functions

Thus, the pulse of Figure 1.9(a) is the sum of the unit step functions of Figures 1.9(b) and 1.9(c) and it is represented as $u_0(t) - u_0(t-1)$.

The unit step function offers a convenient method of describing the sudden application of a voltage or current source. For example, a constant voltage source of 24 V applied at $t = 0$, can be denoted as $24u_0(t)$ V. Likewise, a sinusoidal voltage source $v(t) = V_m \cos\omega t$ V that is applied to a circuit at $t = t_0$, can be described as $v(t) = (V_m \cos\omega t)u_0(t-t_0)$ V. Also, if the excitation in a circuit is a rectangular, or triangular, or sawtooth, or any other recurring pulse, it can be represented as a sum (difference) of unit step functions.

Example 1.2

Express the square waveform of Figure 1.10 as a sum of unit step functions. The vertical dotted lines indicate the discontinuities at T, 2T, 3T, and so on.

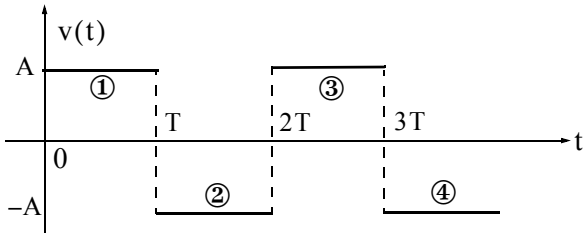


Figure 1.10. Square waveform for Example 1.2

Solution:

Line segment ① has height A, starts at $t = 0$, and terminates at $t = T$. Then, as in Example 1.1, this segment is expressed as

$$v_1(t) = A[u_0(t) - u_0(t-T)] \tag{1.8}$$

Line segment ② has height –A, starts at $t = T$ and terminates at $t = 2T$. This segment is expressed as

$$v_2(t) = -A[u_0(t-T) - u_0(t-2T)] \tag{1.9}$$

Line segment ③ has height A, starts at $t = 2T$ and terminates at $t = 3T$. This segment is expressed as

$$v_3(t) = A[u_0(t-2T) - u_0(t-3T)] \tag{1.10}$$

Line segment ④ has height –A, starts at $t = 3T$, and terminates at $t = 4T$. It is expressed as

$$v_4(t) = -A[u_0(t-3T) - u_0(t-4T)] \tag{1.11}$$

Thus, the square waveform of Figure 1.10 can be expressed as the summation of (1.8) through (1.11), that is,

$$\begin{aligned} v(t) &= v_1(t) + v_2(t) + v_3(t) + v_4(t) \\ &= A[u_0(t) - u_0(t-T)] - A[u_0(t-T) - u_0(t-2T)] \\ &\quad + A[u_0(t-2T) - u_0(t-3T)] - A[u_0(t-3T) - u_0(t-4T)] \end{aligned} \tag{1.12}$$

Combining like terms, we obtain

$$v(t) = A[u_0(t) - 2u_0(t-T) + 2u_0(t-2T) - 2u_0(t-3T) + \ldots] \tag{1.13}$$

Example 1.3

Express the symmetric rectangular pulse of Figure 1.11 as a sum of unit step functions.

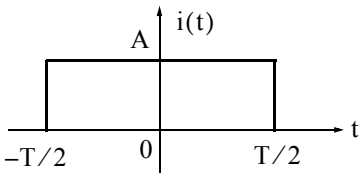


Figure 1.11. Symmetric rectangular pulse for Example 1.3

Solution:

This pulse has height A, starts at $t = -T/2$, and terminates at $t = T/2$. Therefore, with reference to Figures 1.5 and 1.8 (b), we obtain

$$i(t) = Au_0\left(t + \frac{T}{2}\right) - Au_0\left(t - \frac{T}{2}\right) = A\left[u_0\left(t + \frac{T}{2}\right) - u_0\left(t - \frac{T}{2}\right)\right] \tag{1.14}$$

Example 1.4

Express the symmetric triangular waveform of Figure 1.12 as a sum of unit step functions.

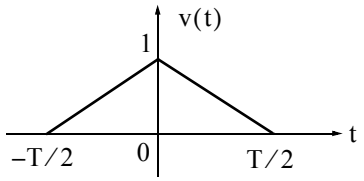


Figure 1.12. Symmetric triangular waveform for Example 1.4

Solution:

We first derive the equations for the linear segments ① and ② shown in Figure 1.13.

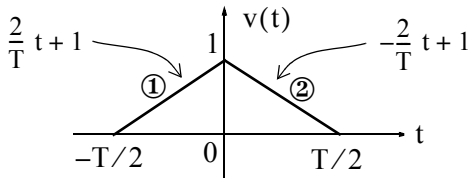


Figure 1.13. Equations for the linear segments of Figure 1.12

For line segment ①,

$$v_1(t) = \left(\frac{2}{T}t + 1\right)\left[u_0\left(t + \frac{T}{2}\right) - u_0(t)\right] \tag{1.15}$$

and for line segment ②,

$$v_2(t) = \left(-\frac{2}{T}t + 1\right)\left[u_0(t) - u_0\left(t - \frac{T}{2}\right)\right] \tag{1.16}$$

Combining (1.15) and (1.16), we obtain

$$\begin{aligned} v(t) &= v_1(t) + v_2(t) \\ &= \left(\frac{2}{T}t + 1\right)\left[u_0\left(t + \frac{T}{2}\right) - u_0(t)\right] + \left(-\frac{2}{T}t + 1\right)\left[u_0(t) - u_0\left(t - \frac{T}{2}\right)\right] \end{aligned} \tag{1.17}$$

Example 1.5

Express the waveform of Figure 1.14 as a sum of unit step functions.

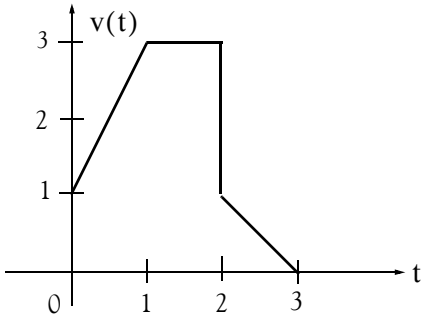


Figure 1.14. Waveform for Example 1.5

Solution:

As in the previous example, we first find the equations of the linear segments linear segments ① and ② shown in Figure 1.15.

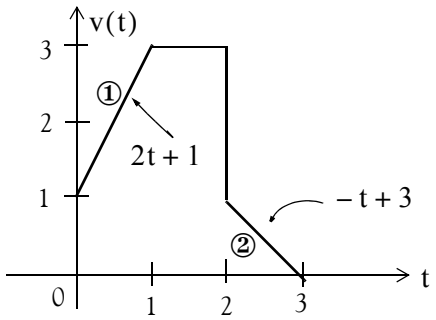


Figure 1.15. Equations for the linear segments of Figure 1.14

Following the same procedure as in the previous examples, we obtain

$$v(t) = (2t+1)[u_0(t) - u_0(t-1)] + 3[u_0(t-1) - u_0(t-2)] + (-t+3)[u_0(t-2) - u_0(t-3)]$$

Multiplying the values in parentheses by the values in the brackets, we obtain

$$v(t) = (2t+1)u_0(t) - (2t+1)u_0(t-1) + 3u_0(t-1) - 3u_0(t-2) + (-t+3)u_0(t-2) - (-t+3)u_0(t-3)$$

$$v(t) = (2t+1)u_0(t) + [-(2t+1)+3]u_0(t-1) + [-3+(-t+3)]u_0(t-2) - (-t+3)u_0(t-3)$$

and combining terms inside the brackets, we obtain

$$v(t) = (2t+1)u_0(t) - 2(t-1)u_0(t-1) - tu_0(t-2) + (t-3)u_0(t-3) \tag{1.18}$$

Two other functions of interest are the *unit ramp function*, and the *unit impulse* or *delta function*. We will introduce them with the examples that follow.

Example 1.6

In the network of Figure 1.16 i_S is a constant current source and the switch is closed at time $t = 0$. Express the capacitor voltage $v_C(t)$ as a function of the unit step.

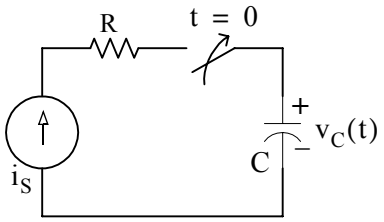


Figure 1.16. Network for Example 1.6

Solution:

The current through the capacitor is $i_C(t) = i_S = \text{constant}$, and the capacitor voltage $v_C(t)$ is

$$v_C(t) = \frac{1}{C}\int_{-\infty}^{t} i_C(\tau)d\tau \; ^* \tag{1.19}$$

where τ is a dummy variable.

Since the switch closes at $t = 0$, we can express the current $i_C(t)$ as

$$i_C(t) = i_S\, u_0(t) \tag{1.20}$$

and assuming that $v_C(t) = 0$ for $t < 0$, we can write (1.19) as

$$v_C(t) = \frac{1}{C}\int_{-\infty}^{t} i_S\, u_0(\tau)d\tau = \underbrace{\frac{i_S}{C}\int_{-\infty}^{0} u_0(\tau)\,d\tau}_{0} + \frac{i_S}{C}\int_{0}^{t} u_0(\tau)d\tau \tag{1.21}$$

or

$$\boxed{v_C(t) = \frac{i_S}{C}\, tu_0(t)} \tag{1.22}$$

Therefore, we see that when a capacitor is charged with a constant current, the voltage across it is a linear function and forms a *ramp* with slope i_S/C as shown in Figure 1.17.

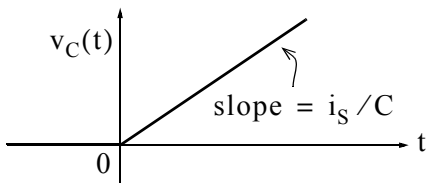


Figure 1.17. Voltage across a capacitor when charged with a constant current source

* *Since the initial condition for the capacitor voltage was not specified, we express this integral with* $-\infty$ *at the lower limit of integration so that any non-zero value prior to* $t < 0$ *would be included in the integration.*

1.3 The Unit Ramp Function $u_1(t)$

The *unit ramp function*, denoted as $u_1(t)$, is defined as

$$u_1(t) = \int_{-\infty}^{t} u_0(\tau)d\tau \tag{1.23}$$

where τ is a dummy variable.

We can evaluate the integral of (1.23) by considering the area under the unit step function $u_0(t)$ from $-\infty$ to t as shown in Figure 1.18.

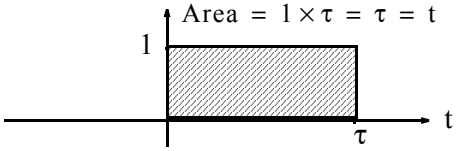


Figure 1.18. Area under the unit step function from $-\infty$ to t

Therefore, we define $u_1(t)$ as

$$u_1(t) = \begin{cases} 0 & t < 0 \\ t & t \geq 0 \end{cases} \tag{1.24}$$

Since $u_1(t)$ is the integral of $u_0(t)$, then $u_0(t)$ must be the derivative of $u_1(t)$, i.e.,

$$\frac{d}{dt}u_1(t) = u_0(t) \tag{1.25}$$

Higher order functions of t can be generated by repeated integration of the unit step function. For example, integrating $u_0(t)$ twice and multiplying by 2, we define $u_2(t)$ as

$$u_2(t) = \begin{cases} 0 & t < 0 \\ t^2 & t \geq 0 \end{cases} \quad \text{or} \quad u_2(t) = 2\int_{-\infty}^{t} u_1(\tau)d\tau \tag{1.26}$$

Similarly,

$$u_3(t) = \begin{cases} 0 & t < 0 \\ t^3 & t \geq 0 \end{cases} \quad \text{or} \quad u_3(t) = 3\int_{-\infty}^{t} u_2(\tau)d\tau \tag{1.27}$$

and in general,

$$u_n(t) = \begin{cases} 0 & t < 0 \\ t^n & t \geq 0 \end{cases} \quad \text{or} \quad u_n(t) = 3\int_{-\infty}^{t} u_{n-1}(\tau)d\tau \tag{1.28}$$

Also,

$$u_{n-1}(t) = \frac{1}{n}\frac{d}{dt}u_n(t) \tag{1.29}$$

Example 1.7

In the network of Figure 1.19, the switch is closed at time $t = 0$ and $i_L(t) = 0$ for $t < 0$. Express the inductor current $i_L(t)$ in terms of the unit step function.

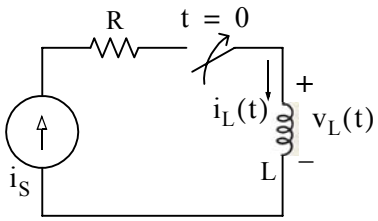


Figure 1.19. Network for Example 1.7

Solution:

The voltage across the inductor is

$$v_L(t) = L\frac{di_L}{dt} \tag{1.30}$$

and since the switch closes at $t = 0$,

$$i_L(t) = i_S\, u_0(t) \tag{1.31}$$

Therefore, we can write (1.30) as

$$\boxed{v_L(t) = Li_S\frac{d}{dt}u_0(t)} \tag{1.32}$$

But, as we know, $u_0(t)$ is constant (0 or 1) for all time except at $t = 0$ where it is discontinuous. Since the derivative of any constant is zero, the derivative of the unit step $u_0(t)$ has a non–zero value only at $t = 0$. The derivative of the unit step function is defined in the next section.

1.4 The Delta Function $\delta(t)$

The *unit impulse* or *delta function*, denoted as $\delta(t)$, is the derivative of the unit step $u_0(t)$. It is also defined as

$$\boxed{\int_{-\infty}^{t}\delta(\tau)d\tau = u_0(t)} \tag{1.33}$$

and

$$\boxed{\delta(t) = 0 \text{ for all } t \neq 0} \tag{1.34}$$

To better understand the delta function $\delta(t)$, let us represent the unit step $u_0(t)$ as shown in Figure 1.20 (a).

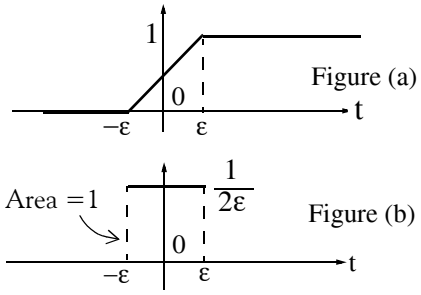


Figure 1.20. Representation of the unit step as a limit

The function of Figure 1.20 (a) becomes the unit step as $\varepsilon \to 0$. Figure 1.20 (b) is the derivative of Figure 1.20 (a), where we see that as $\varepsilon \to 0$, $1/2\varepsilon$ becomes unbounded, but the area of the rectangle remains 1. Therefore, in the limit, we can think of $\delta(t)$ as approaching a very large spike or impulse at the origin, with unbounded amplitude, zero width, and area equal to 1.

Two useful properties of the delta function are the sampling property and the sifting property.

1.4.1 The Sampling Property of the Delta Function $\delta(t)$

The *sampling property* of the delta function states that

$$\boxed{f(t)\delta(t-a) = f(a)\delta(t)} \tag{1.35}$$

or, when $a = 0$,

$$\boxed{f(t)\delta(t) = f(0)\delta(t)} \tag{1.36}$$

that is, multiplication of any function $f(t)$ by the delta function $\delta(t)$ results in sampling the function at the time instants where the delta function is not zero. The study of discrete–time systems is based on this property.

Proof:

Since $\delta(t) = 0$ for $t<0$ and $t>0$ then,

$$f(t)\delta(t) = 0 \quad \text{for} \quad t<0 \quad \text{and} \quad t>0 \tag{1.37}$$

We rewrite $f(t)$ as

$$f(t) = f(0) + [f(t) - f(0)] \tag{1.38}$$

Integrating (1.37) over the interval $-\infty$ to t and using (1.38), we obtain

$$\int_{-\infty}^{t} f(\tau)\delta(\tau)d\tau = \int_{-\infty}^{t} f(0)\delta(\tau)d\tau + \int_{-\infty}^{t} [f(\tau) - f(0)]\delta(\tau)d\tau \tag{1.39}$$

The first integral on the right side of (1.39) contains the constant term $f(0)$; this can be written outside the integral, that is,

$$\int_{-\infty}^{t} f(0)\delta(\tau)d\tau = f(0)\int_{-\infty}^{t} \delta(\tau)d\tau \tag{1.40}$$

The second integral of the right side of (1.39) is always zero because

$$\delta(t) = 0 \quad \text{for} \quad t < 0 \quad \text{and} \quad t > 0$$

and

$$[f(\tau) - f(0)]\big|_{\tau = 0} = f(0) - f(0) = 0$$

Therefore, (1.39) reduces to

$$\int_{-\infty}^{t} f(\tau)\delta(\tau)d\tau = f(0)\int_{-\infty}^{t} \delta(\tau)d\tau \tag{1.41}$$

Differentiating both sides of (1.41), and replacing τ with t, we obtain

$$\boxed{\begin{array}{c} f(t)\delta(t) = f(0)\delta(t) \\ \text{Sampling Property of } \delta(t) \end{array}} \tag{1.42}$$

1.4.2 The Sifting Property of the Delta Function $\delta(t)$

The *sifting property* of the delta function states that

$$\boxed{\int_{-\infty}^{\infty} f(t)\delta(t-\alpha)dt = f(\alpha)} \tag{1.43}$$

that is, if we multiply any function $f(t)$ by $\delta(t-\alpha)$, and integrate from $-\infty$ to $+\infty$, we will obtain the value of $f(t)$ evaluated at $t = \alpha$.

Proof:

Let us consider the integral

$$\int_{a}^{b} f(t)\delta(t-\alpha)dt \quad \text{where} \quad a < \alpha < b \tag{1.44}$$

We will use integration by parts to evaluate this integral. We recall from the derivative of products that

$$d(xy) = xdy + ydx \quad \text{or} \quad xdy = d(xy) - ydx \tag{1.45}$$

and integrating both sides we obtain

$$\int x\,dy = xy - \int y\,dx \tag{1.46}$$

Now, we let $x = f(t)$; then, $dx = f'(t)$. We also let $dy = \delta(t-\alpha)$; then, $y = u_0(t-\alpha)$. By substitution into (1.44), we obtain

$$\int_a^b f(t)\delta(t-\alpha)dt = f(t)u_0(t-\alpha)\Big|_a^b - \int_a^b u_0(t-\alpha)f'(t)dt \tag{1.47}$$

We have assumed that $a<\alpha<b$; therefore, $u_0(t-\alpha) = 0$ for $\alpha<a$, and thus the first term of the right side of (1.47) reduces to $f(b)$. Also, the integral on the right side is zero for $\alpha<a$, and therefore, we can replace the lower limit of integration a by α. We can now rewrite (1.47) as

$$\int_a^b f(t)\delta(t-\alpha)dt = f(b) - \int_\alpha^b f'(t)dt = f(b) - f(b) + f(\alpha)$$

and letting $a \to -\infty$ and $b \to \infty$ for any $|\alpha| < \infty$, we obtain

$$\boxed{\int_{-\infty}^{\infty} f(t)\delta(t-\alpha)dt = f(\alpha)}$$
$$\text{Sifting Property of } \delta(t) \tag{1.48}$$

1.5 Higher Order Delta Functions

An *nth-order delta function* is defined as the nth derivative of $u_0(t)$, that is,

$$\boxed{\delta^n(t) = \frac{\delta^n}{dt}[u_0(t)]} \tag{1.49}$$

The function $\delta'(t)$ is called *doublet*, $\delta''(t)$ is called *triplet*, and so on. By a procedure similar to the derivation of the sampling property of the delta function, we can show that

$$\boxed{f(t)\delta'(t-a) = f(a)\delta'(t-a) - f'(a)\delta(t-a)} \tag{1.50}$$

Also, the derivation of the sifting property of the delta function can be extended to show that

$$\boxed{\int_{-\infty}^{\infty} f(t)\delta^n(t-\alpha)dt = (-1)^n \frac{d^n}{dt^n}[f(t)]\Big|_{t=\alpha}} \tag{1.51}$$

Example 1.8

Evaluate the following expressions:

a. $3t^4\delta(t-1)$ b. $\int_{-\infty}^{\infty} t\delta(t-2)dt$ c. $t^2\delta'(t-3)$

Solution:

a. The sampling property states that $f(t)\delta(t-a) = f(a)\delta(t-a)$ For this example, $f(t) = 3t^4$ and $a = 1$. Then,

$$3t^4\delta(t-1) = \{3t^4\big|_{t=1}\}\delta(t-1) = 3\delta(t-1)$$

b. The sifting property states that $\int_{-\infty}^{\infty} f(t)\delta(t-\alpha)dt = f(\alpha)$. For this example, $f(t) = t$ and $\alpha = 2$. Then,

$$\int_{-\infty}^{\infty} t\delta(t-2)dt = f(2) = t\big|_{t=2} = 2$$

c. The given expression contains the doublet; therefore, we use the relation

$$f(t)\delta'(t-a) = f(a)\delta'(t-a) - f'(a)\delta(t-a)$$

Then, for this example,

$$t^2\delta'(t-3) = t^2\big|_{t=3}\delta'(t-3) - \frac{d}{dt}t^2\big|_{t=3}\delta(t-3) = 9\delta'(t-3) - 6\delta(t-3)$$

Example 1.9

a. Express the voltage waveform $v(t)$ shown in Figure 1.21 as a sum of unit step functions for the time interval $-1 < t < 7$ s.

b. Using the result of part (a), compute the derivative of $v(t)$ and sketch its waveform.

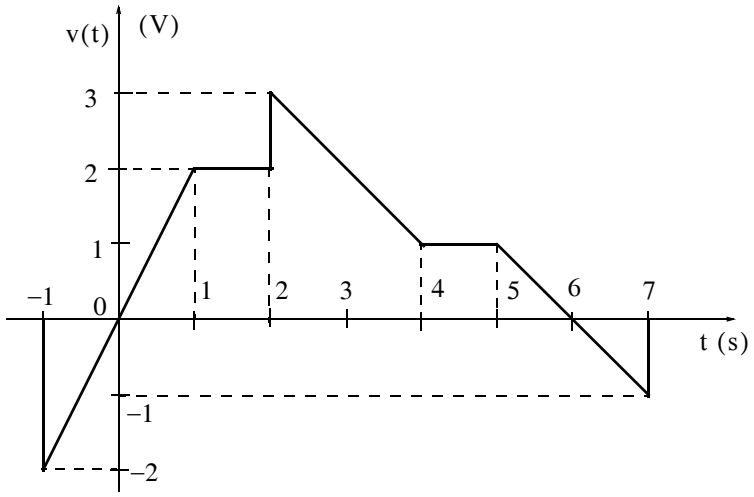


Figure 1.21. Waveform for Example 1.9

Solution:

a. We begin with the derivation of the equations for the linear segments of the given waveform as shown in Figure 1.22.

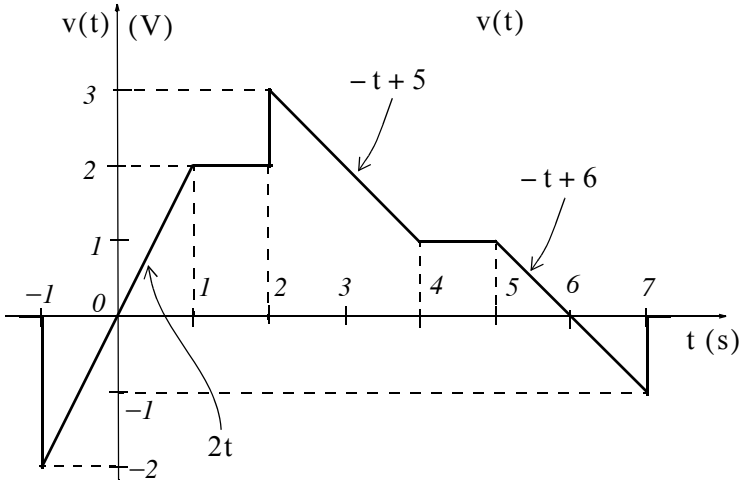


Figure 1.22. Equations for the linear segments of Figure 1.21

Next, we express $v(t)$ in terms of the unit step function $u_0(t)$, and we obtain

$$\begin{aligned} v(t) &= 2t[u_0(t+1)-u_0(t-1)]+2[u_0(t-1)-u_0(t-2)] \\ &\quad +(-t+5)[u_0(t-2)-u_0(t-4)]+[u_0(t-4)-u_0(t-5)] \\ &\quad +(-t+6)[u_0(t-5)-u_0(t-7)] \end{aligned} \tag{1.52}$$

Multiplying and collecting like terms in (1.52), we obtain

$$\begin{aligned} v(t) &= 2tu_0(t+1) - 2tu_0(t-1) - 2u_0(t-1) - 2u_0(t-2) - tu_0(t-2) \\ &\quad + 5u_0(t-2) + tu_0(t-4) - 5u_0(t-4) + u_0(t-4) - u_0(t-5) \\ &\quad -tu_0(t-5) + 6u_0(t-5) + tu_0(t-7) - 6u_0(t-7) \end{aligned}$$

or

$$\begin{aligned} v(t) &= 2tu_0(t+1) + (-2t+2)u_0(t-1) + (-t+3)u_0(t-2) \\ &\quad + (t-4)u_0(t-4) + (-t+5)u_0(t-5) + (t-6)u_0(t-7) \end{aligned}$$

b. The derivative of $v(t)$ is

$$\begin{aligned} \frac{dv}{dt} &= 2u_0(t+1) + 2t\delta(t+1) - 2u_0(t-1) + (-2t+2)\delta(t-1) \\ &\quad - u_0(t-2) + (-t+3)\delta(t-2) + u_0(t-4) + (t-4)\delta(t-4) \\ &\quad - u_0(t-5) + (-t+5)\delta(t-5) + u_0(t-7) + (t-6)\delta(t-7) \end{aligned} \tag{1.53}$$

From the given waveform, we observe that discontinuities occur only at $t = -1$, $t = 2$, and $t = 7$. Therefore, $\delta(t-1) = 0$, $\delta(t-4) = 0$, and $\delta(t-5) = 0$, and the terms that contain these delta functions vanish. Also, by application of the sampling property,

$$\begin{aligned} 2t\delta(t+1) &= \{2t|_{t=-1}\}\delta(t+1) = -2\delta(t+1) \\ (-t+3)\delta(t-2) &= \{(-t+3)|_{t=2}\}\delta(t-2) = \delta(t-2) \\ (t-6)\delta(t-7) &= \{(t-6)|_{t=7}\}\delta(t-7) = \delta(t-7) \end{aligned}$$

and by substitution into (1.53), we obtain

$$\begin{aligned} \frac{dv}{dt} &= 2u_0(t+1) - 2\delta(t+1) - 2u_0(t-1) - u_0(t-2) \\ &\quad + \delta(t-2) + u_0(t-4) - u_0(t-5) + u_0(t-7) + \delta(t-7) \end{aligned} \tag{1.54}$$

The plot of dv/dt is shown in Figure 1.23.

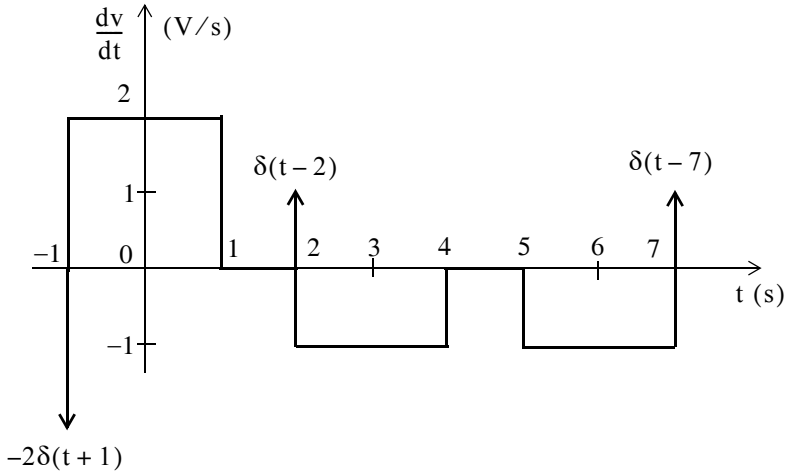


Figure 1.23. Plot of the derivative of the waveform of Figure 1.21

We observe that a negative spike of magnitude 2 occurs at $t = -1$, and two positive spikes of magnitude 1 occur at $t = 2$, and $t = 7$. These spikes occur because of the discontinuities at these points.

It would be interesting to observe the given signal and its derivative on the Scope block of the Simulink®* model of Figure 1.24. They are shown in Figure 1.25.

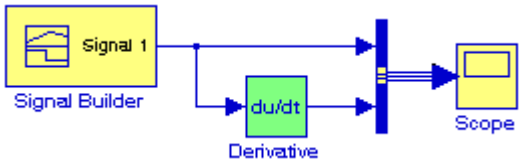


Figure 1.24. Simulink model for Example 1.9

The waveform created by the Signal Builder block is shown in Figure 1.25.

* *A brief introduction to Simulink is presented in Appendix B. For a detailed procedure for generating piece-wise linear functions with Simulink's Signal Builder block, please refer to Introduction to Simulink with Engineering Applications, ISBN 0–9744239–7–1*

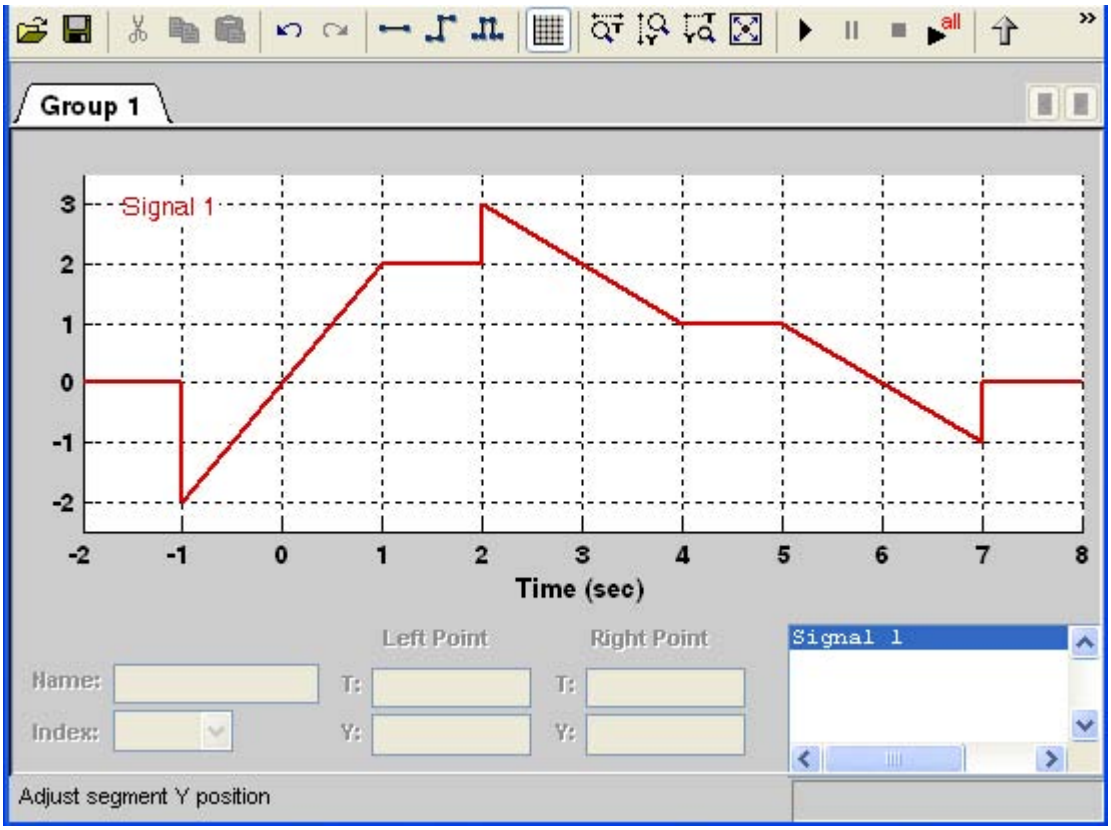


Figure 1.25. Piece–wise linear waveform for the Signal Builder block in Figure 1.24

The waveform in Figure 1.25 is created with the following procedure:

1. We open a new model by clicking on the new model icon shown as a blank page on the left corner of the top menu bar. Initially, the name **Untitled** appears on the top of this new model. We save it with the name Figure_1.25 and Simulink appends the .mdl extension to it.

2. From the **Sources** library, we drag the **Signal Builder** block into this new model. We also drag the **Derivative** block from the **Continuous** library, the **Bus Creator** block from the **Commonly Used Blocks** library, and the **Scope** block into this model, and we interconnect these blocks as shown in Figure 1.24.

3. We double–click on the Signal Builder block in Figure 1.24, and on the plot which appears as a square pulse, we click on the y–axis and we enter Minimum: –2.5, and Maximum: 3.5. Likewise we right–click anywhere on the plot and we specify the Change Time Range at Min time: –2, and Max time: 8.

4. To select a particular point, we position the mouse cursor over that point and we left–click. A circle is drawn around that point to indicate that it is selected.

5. To select a line segment, we left–click on that segment. That line segment is now shown as a thick line indicating that it is selected. To deselect it, we press the Esc key.

6. To drag a line segment to a new position, we place the mouse cursor over that line segment and the cursor shape shows the position in which we can drag the segment.
7. To drag a point along the y–axis, we move the mouse cursor over that point, and the cursor changes to a circle indicating that we can drag that point. Then, we can move that point in a direction parallel to the x–axis.
8. To drag a point along the x–axis, we select that point, and we hold down the Shift key while dragging that point.
9. When we select a line segment on the time axis (x–axis) we observe that at the lower end of the waveform display window the **Left Point** and **Right Point** fields become visible. We can then reshape the given waveform by specifying the Time (**T**) and Amplitude (**Y**) points.

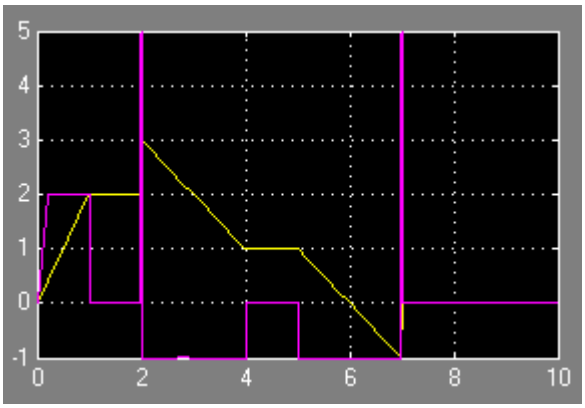


Figure 1.26. Waveforms for the Simulink model of Figure 1.24

The two positive spikes that occur at $t = 2$, and $t = 7$, are clearly shown in Figure 1.26.

MATLAB* has built-in functions for the unit step, and the delta functions. These are denoted by the names of the mathematicians who used them in their work. The unit step function $u_0(t)$ is referred to as **Heaviside(t)**, and the delta function $\delta(t)$ is referred to as **Dirac(t)**. Their use is illustrated with the examples below.

```
syms k a t;                          % Define symbolic variables
u=k*sym('Heaviside(t–a)')            % Create unit step function at t = a

u =
k*Heaviside(t-a)

d=diff(u)                            % Compute the derivative of the unit step function

d =
k*Dirac(t-a)
```

* *An introduction to MATLAB® is given in Appendix A.*

```
int(d)                          % Integrate the delta function
ans =
Heaviside(t-a)*k
```

1.6 Summary

- The unit step function $u_0(t)$ is defined as

$$u_0(t) = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \end{cases}$$

- The unit step function offers a convenient method of describing the sudden application of a voltage or current source.
- The unit ramp function, denoted as $u_1(t)$, is defined as

$$u_1(t) = \int_{-\infty}^{t} u_0(\tau)d\tau$$

- The unit impulse or delta function, denoted as $\delta(t)$, is the derivative of the unit step $u_0(t)$. It is also defined as

$$\int_{-\infty}^{t} \delta(\tau)d\tau = u_0(t)$$

and

$$\delta(t) = 0 \quad \text{for all} \quad t \neq 0$$

- The sampling property of the delta function states that

$$f(t)\delta(t-a) = f(a)\delta(t)$$

or, when $a = 0$,

$$f(t)\delta(t) = f(0)\delta(t)$$

- The sifting property of the delta function states that

$$\int_{-\infty}^{\infty} f(t)\delta(t-\alpha)dt = f(\alpha)$$

- The sampling property of the doublet function $\delta'(t)$ states that

$$f(t)\delta'(t-a) = f(a)\delta'(t-a) - f'(a)\delta(t-a)$$

1.7 Exercises

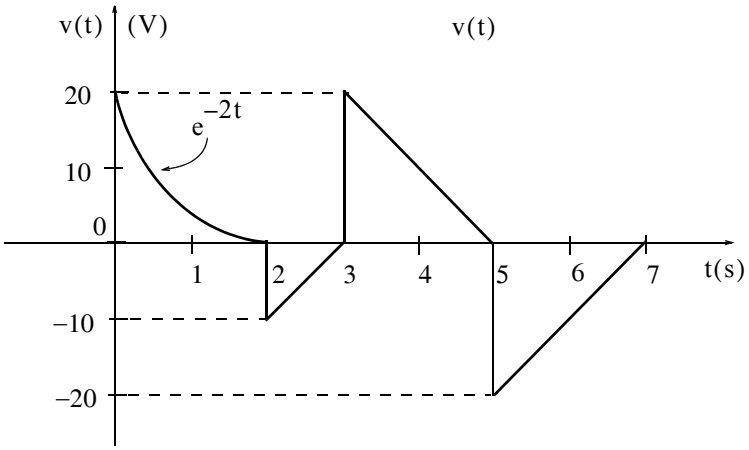
1. Evaluate the following functions:

a. $\sin t\delta\left(t-\frac{\pi}{6}\right)$ b. $\cos 2t\delta\left(t-\frac{\pi}{4}\right)$ c. $\cos^2 t\delta\left(t-\frac{\pi}{2}\right)$

d. $\tan 2t\delta\left(t-\frac{\pi}{8}\right)$ e. $\int_{-\infty}^{\infty} t^2 e^{-t}\delta(t-2)dt$ f. $\sin^2 t\delta^1\left(t-\frac{\pi}{2}\right)$

2.

a. Express the voltage waveform $v(t)$ shown below as a sum of unit step functions for the time interval $0 < t < 7$ s.



b. Using the result of part (a), compute the derivative of $v(t)$, and sketch its waveform. This waveform cannot be used with Sinulink's Function Builder block because it contains the decaying exponential segment which is a non–linear function.

1.8 Solutions to End–of–Chapter Exercises

Dear Reader:

The remaining pages on this chapter contain the solutions to the exercises.

You must, for your benefit, make an honest effort to solve the problems without first looking at the solutions that follow. It is recommended that first you go through and solve those you feel that you know. For the exercises that you are uncertain, review this chapter and try again. If your results do not agree with those provided, look over your procedures for inconsistencies and computational errors. Refer to the solutions as a last resort and rework those problems at a later date.

You should follow this practice with the exercises on all chapters of this book.

1. We apply the sampling property of the $\delta(t)$ function for all expressions except (e) where we apply the sifting property. For part (f) we apply the sampling property of the doublet.

We recall that the sampling property states that $f(t)\delta(t-a) = f(a)\delta(t-a)$. Thus,

a. $\sin t\delta\left(t-\frac{\pi}{6}\right) = \sin t|_{t=\pi/6}\delta\left(t-\frac{\pi}{6}\right) = \sin\frac{\pi}{6}\delta\left(t-\frac{\pi}{6}\right) = 0.5\delta\left(t-\frac{\pi}{6}\right)$

b. $\cos 2t\delta\left(t-\frac{\pi}{4}\right) = \cos 2t|_{t=\pi/4}\delta\left(t-\frac{\pi}{4}\right) = \cos\frac{\pi}{2}\delta\left(t-\frac{\pi}{4}\right) = 0$

c. $\cos^2 t\delta\left(t-\frac{\pi}{2}\right) = \frac{1}{2}(1+\cos 2t)\Big|_{t=\pi/2}\delta\left(t-\frac{\pi}{2}\right) = \frac{1}{2}(1+\cos\pi)\delta\left(t-\frac{\pi}{2}\right) = \frac{1}{2}(1-1)\delta\left(t-\frac{\pi}{2}\right) = 0$

d. $\tan 2t\delta\left(t-\frac{\pi}{8}\right) = \tan 2t|_{t=\pi/8}\delta\left(t-\frac{\pi}{8}\right) = \tan\frac{\pi}{4}\delta\left(t-\frac{\pi}{8}\right) = \delta\left(t-\frac{\pi}{8}\right)$

We recall that the sampling property states that $\int_{-\infty}^{\infty} f(t)\delta(t-\alpha)dt = f(\alpha)$. Thus,

e. $\int_{-\infty}^{\infty} t^2e^{-t}\delta(t-2)dt = t^2e^{-t}\Big|_{t=2} = 4e^{-2} = 0.54$

We recall that the sampling property for the doublet states that

$$f(t)\delta'(t-a) = f(a)\delta'(t-a) - f'(a)\delta(t-a)$$

Thus,

$$\sin^2 t\delta^1\left(t-\frac{\pi}{2}\right) = \sin^2 t|_{t=\pi/2}\delta^1\left(t-\frac{\pi}{2}\right) - \frac{d}{dt}\sin^2 t|_{t=\pi/2}\delta\left(t-\frac{\pi}{2}\right)$$

f.
$$= \frac{1}{2}(1-\cos 2t)|_{t=\pi/2}\delta^1\left(t-\frac{\pi}{2}\right) - \sin 2t|_{t=\pi/2}\delta\left(t-\frac{\pi}{2}\right)$$

$$= \frac{1}{2}(1+1)\delta^1\left(t-\frac{\pi}{2}\right) - \sin\pi\delta\left(t-\frac{\pi}{2}\right) = \delta^1\left(t-\frac{\pi}{2}\right)$$

2.

a.
$$v(t) = e^{-2t}[u_0(t)-u_0(t-2)] + (10t-30)[u_0(t-2)-u_0(t-3)] + (-10t+50)[u_0(t-3)-u_0(t-5)] + (10t-70)[u_0(t-5)-u_0(t-7)]$$

$$v(t) = e^{-2t}u_0(t) - e^{-2t}u_0(t-2) + 10tu_0(t-2) - 30u_0(t-2) - 10tu_0(t-3) + 30u_0(t-3) - 10tu_0(t-3) + 50u_0(t-3) + 10tu_0(t-5) - 50u_0(t-5) + 10tu_0(t-5) - 70u_0(t-5) - 10tu_0(t-7) + 70u_0(t-7)$$

$$v(t) = e^{-2t}u_0(t) + (-e^{-2t} + 10t - 30)u_0(t-2) + (-20t + 80)u_0(t-3) + (20t - 120)u_0(t-5)$$
$$+(-10t + 70)u_0(t-7)$$

b.

$$\frac{dv}{dt} = -2e^{-2t}u_0(t) + e^{-2t}\delta(t) + (2e^{-2t} + 10)u_0(t-2) + (-e^{-2t} + 10t - 30)\delta(t-2)$$
$$-20u_0(t-3) + (-20t + 80)\delta(t-3) + 20u_0(t-5) + (20t - 120)\delta(t-5) \qquad (1)$$
$$-10u_0(t-7) + (-10t + 70)\delta(t-7)$$

Referring to the given waveform we observe that discontinuities occur only at $t = 2$, $t = 3$, and $t = 5$. Therefore, $\delta(t) = 0$ and $\delta(t-7) = 0$. Also, by the sampling property of the delta function

$$(-e^{-2t} + 10t - 30)\delta(t-2) = (-e^{-2t} + 10t - 30)\Big|_{t=2}\delta(t-2) \approx -10\delta(t-2)$$

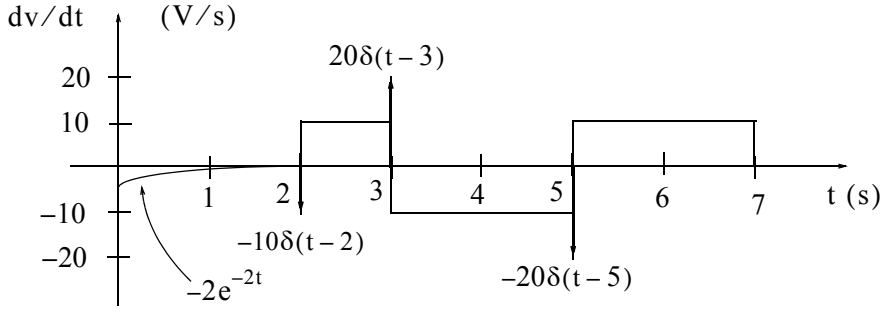
$$(-20t + 80)\delta(t-3) = (-20t + 80)\Big|_{t=3}\delta(t-3) = 20\delta(t-3)$$

$$(20t - 120)\delta(t-5) = (20t - 120)\Big|_{t=5}\delta(t-5) = -20\delta(t-5)$$

and with these simplifications (1) above reduces to

$$dv/dt = -2e^{-2t}u_0(t) + 2e^{-2t}u_0(t-2) + 10u_0(t-2) - 10\delta(t-2)$$
$$-20u_0(t-3) + 20\delta(t-3) + 20u_0(t-5) - 20\delta(t-5) - 10u_0(t-7)$$
$$= -2e^{-2t}[u_0(t) - u_0(t-2)] - 10\delta(t-2) + 10[u_0(t-2) - u_0(t-3)] + 20\delta(t-3)$$
$$- 10[u_0(t-3) - u_0(t-5)] - 20\delta(t-5) + 10[u_0(t-5) - u_0(t-7)]$$

The waveform for dv/dt is shown below.



Chapter 2

The Laplace Transformation

This chapter begins with an introduction to the Laplace transformation, definitions, and properties of the Laplace transformation. The initial value and final value theorems are also discussed and proved. It continues with the derivation of the Laplace transform of common functions of time, and concludes with the derivation of the Laplace transforms of common waveforms.

2.1 Definition of the Laplace Transformation

The *two–sided* or *bilateral* Laplace Transform pair is defined as

$$\mathcal{L}\{f(t)\} = F(s) = \int_{-\infty}^{\infty} f(t)e^{-st}dt \tag{2.1}$$

$$\mathcal{L}^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F(s)e^{st}ds \tag{2.2}$$

where $\mathcal{L}\{f(t)\}$ denotes the Laplace transform of the time function $f(t)$, $\mathcal{L}^{-1}\{F(s)\}$ denotes the Inverse Laplace transform, and s is a complex variable whose real part is σ, and imaginary part ω, that is, $s = \sigma + j\omega$.

In most problems, we are concerned with values of time t greater than some reference time, say $t = t_0 = 0$, and since the initial conditions are generally known, the two–sided Laplace transform pair of (2.1) and (2.2) simplifies to the *unilateral* or *one–sided Laplace transform* defined as

$$\mathcal{L}\{f(t)\} = F(s) = \int_{t_0}^{\infty} f(t)e^{-st}dt = \int_{0}^{\infty} f(t)e^{-st}dt \tag{2.3}$$

$$\mathcal{L}^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F(s)e^{st}ds \tag{2.4}$$

The Laplace Transform of (2.3) has meaning only if the integral converges (reaches a limit), that is, if

$$\left|\int_0^\infty f(t)e^{-st}dt\right| < \infty \tag{2.5}$$

To determine the conditions that will ensure us that the integral of (2.3) converges, we rewrite (2.5) as

$$\left|\int_0^\infty f(t)e^{-\sigma t}e^{-j\omega t}dt\right| < \infty \tag{2.6}$$

The term $e^{-j\omega t}$ in the integral of (2.6) has magnitude of unity, i.e., $\left|e^{-j\omega t}\right| = 1$, and thus the condition for convergence becomes

$$\left|\int_0^\infty f(t)e^{-\sigma t}dt\right| < \infty \tag{2.7}$$

Fortunately, in most engineering applications the functions $f(t)$ are of *exponential order**. Then, we can express (2.7) as,

$$\left|\int_0^\infty f(t)e^{-\sigma t}dt\right| < \left|\int_0^\infty ke^{\sigma_0 t}e^{-\sigma t}dt\right| \tag{2.8}$$

and we see that the integral on the right side of the inequality sign in (2.8), converges if $\sigma > \sigma_0$. Therefore, we conclude that if $f(t)$ is of exponential order, $\mathcal{L}\{f(t)\}$ exists if

$$\mathrm{Re}\{s\} = \sigma > \sigma_0 \tag{2.9}$$

where $\mathrm{Re}\{s\}$ denotes the real part of the complex variable s.

Evaluation of the integral of (2.4) involves contour integration in the complex plane, and thus, it will not be attempted in this chapter. We will see in the next chapter that many Laplace transforms can be inverted with the use of a few standard pairs, and thus there is no need to use (2.4) to obtain the Inverse Laplace transform.

In our subsequent discussion, we will denote transformation from the time domain to the complex frequency domain, and vice versa, as

$$f(t) \Leftrightarrow F(s) \tag{2.10}$$

2.2 Properties and Theorems of the Laplace Transform

The most common properties and theorems of the Laplace transform are presented in Subsections 2.2.1 through 2.2.13 below.

---

* *A function* $f(t)$ *is said to be of exponential order if* $|f(t)| < ke^{\sigma_0 t}$ *for all* $t \geq 0$.

2.2.1 Linearity Property

The *linearity property* states that if

$$f_1(t),\ f_2(t),\ \ldots,\ f_n(t)$$

have Laplace transforms

$$F_1(s),\ F_2(s),\ \ldots,\ F_n(s)$$

respectively, and

$$c_1,\ c_2,\ \ldots,\ c_n$$

are arbitrary constants, then,

$$\boxed{c_1\,f_1(t)+c_2\,f_2(t)+\ldots+c_n\,f_n(t)\Leftrightarrow c_1\,F_1(s)+c_2\,F_2(s)+\ldots+c_n\,F_n(s)} \tag{2.11}$$

Proof:

$$\begin{aligned}\mathscr{L}\{c_1\,f_1(t)+c_2\,f_2(t)+\ldots+c_n\,f_n(t)\} &= \int_{t_0}^{\infty}[c_1\,f_1(t)+c_2\,f_2(t)+\ldots+c_n\,f_n(t)]dt\\ &= c_1\int_{t_0}^{\infty}f_1(t)e^{-st}dt+c_2\int_{t_0}^{\infty}f_2(t)e^{-st}dt+\ldots+c_n\int_{t_0}^{\infty}f_n(t)e^{-st}dt\\ &= c_1\,F_1(s)+c_2\,F_2(s)+\ldots+c_n\,F_n(s)\end{aligned}$$

Note 1:

It is desirable to multiply $f(t)$ by the unit step function $u_0(t)$ to eliminate any unwanted non–zero values of $f(t)$ for $t<0$.

2.2.2 Time Shifting Property

The *time shifting property* states that a right shift in the time domain by a units, corresponds to multiplication by e^{-as} in the complex frequency domain. Thus,

$$\boxed{f(t-a)u_0(t-a)\Leftrightarrow e^{-as}F(s)} \tag{2.12}$$

Proof:

$$\mathscr{L}\{f(t-a)u_0(t-a)\} = \int_0^a 0e^{-st}dt+\int_a^{\infty}f(t-a)e^{-st}dt \tag{2.13}$$

Now, we let $t-a=\tau$; then, $t=\tau+a$ and $dt=d\tau$. With these substitutions and with $a\to 0$, the second integral on the right side of (2.13) is expressed as

$$\int_0^{\infty}f(\tau)e^{-s(\tau+a)}d\tau = e^{-as}\int_0^{\infty}f(\tau)e^{-s\tau}d\tau = e^{-as}F(s)$$

2.2.3 Frequency Shifting Property

The *frequency shifting property* states that if we multiply a time domain function $f(t)$ by an exponential function e^{-at} where a is an arbitrary positive constant, this multiplication will produce a shift of the s variable in the complex frequency domain by a units. Thus,

$$\boxed{e^{-at}f(t) \Leftrightarrow F(s+a)} \tag{2.14}$$

Proof:

$$\mathcal{L}\{e^{-at}f(t)\} = \int_0^\infty e^{-at}f(t)e^{-st}dt = \int_0^\infty f(t)e^{-(s+a)t}dt = F(s+a)$$

Note 2:

A change of scale is represented by multiplication of the time variable t by a positive scaling factor a. Thus, the function $f(t)$ after scaling the time axis, becomes $f(at)$.

2.2.4 Scaling Property

Let a be an arbitrary positive constant; then, the *scaling property* states that

$$\boxed{f(at) \Leftrightarrow \frac{1}{a}F\left(\frac{s}{a}\right)} \tag{2.15}$$

Proof:

$$\mathcal{L}\{f(at)\} = \int_0^\infty f(at)e^{-st}dt$$

and letting $t = \tau/a$, we obtain

$$\mathcal{L}\{f(at)\} = \int_0^\infty f(\tau)e^{-s(\tau/a)}d\left(\frac{\tau}{a}\right) = \frac{1}{a}\int_0^\infty f(\tau)e^{-(s/a)\tau}d(\tau) = \frac{1}{a}F\left(\frac{s}{a}\right)$$

Note 3:

Generally, the initial value of $f(t)$ is taken at $t = 0^-$ to include any discontinuity that may be present at $t = 0$. If it is known that no such discontinuity exists at $t = 0^-$, we simply interpret $f(0^-)$ as $f(0)$.

2.2.5 Differentiation in Time Domain Property

The *differentiation in time domain* property states that differentiation in the time domain corresponds to multiplication by s in the complex frequency domain, minus the initial value of $f(t)$ at $t = 0^-$. Thus,

$$\boxed{f'(t) = \frac{d}{dt} f(t) \Leftrightarrow sF(s) - f(0^-)} \tag{2.16}$$

Proof:

$$\mathcal{L}\{f'(t)\} = \int_0^\infty f'(t)e^{-st}dt$$

Using integration by parts where

$$\int vdu = uv - \int udv \tag{2.17}$$

we let $du = f'(t)$ and $v = e^{-st}$. Then, $u = f(t)$, $dv = -se^{-st}$, and thus

$$\mathcal{L}\{f'(t)\} = f(t)e^{-st}\Big|_{0^-}^{\infty} + s\int_{0^-}^{\infty} f(t)e^{-st}dt = \lim_{a\to\infty}\left[f(t)e^{-st}\Big|_{0^-}^{a}\right] + sF(s)$$

$$= \lim_{a\to\infty}[e^{-sa}f(a) - f(0^-)] + sF(s) = 0 - f(0^-) + sF(s)$$

The time differentiation property can be extended to show that

$$\boxed{\frac{d^2}{dt^2} f(t) \Leftrightarrow s^2F(s) - sf(0^-) - f'(0^-)} \tag{2.18}$$

$$\boxed{\frac{d^3}{dt^3} f(t) \Leftrightarrow s^3F(s) - s^2f(0^-) - sf'(0^-) - f''(0^-)} \tag{2.19}$$

and in general

$$\boxed{\frac{d^n}{dt^n} f(t) \Leftrightarrow s^nF(s) - s^{n-1}f(0^-) - s^{n-2}f'(0^-) - \ldots - f^{n-1}(0^-)} \tag{2.20}$$

To prove (2.18), we let

$$g(t) = f'(t) = \frac{d}{dt} f(t)$$

and as we found above,

$$\mathcal{L}\{g'(t)\} = s\mathcal{L}\{g(t)\} - g(0^-)$$

Then,

$$\mathcal{L}\{f''(t)\} = s\mathcal{L}\{f'(t)\} - f'(0^-) = s[s\mathcal{L}[f(t)] - f(0^-)] - f'(0^-)$$

$$= s^2F(s) - sf(0^-) - f'(0^-)$$

Relations (2.19) and (2.20) can be proved by similar procedures.

We must remember that the terms $f(0^-)$, $f'(0^-)$, $f''(0^-)$, and so on, represent the initial conditions. Therefore, when all initial conditions are zero, and we differentiate a time function $f(t)$ n times, this corresponds to $F(s)$ multiplied by s to the nth power.

2.2.6 Differentiation in Complex Frequency Domain Property

This property states that *differentiation in complex frequency domain* and multiplication by minus one, corresponds to multiplication of $f(t)$ by t in the time domain. In other words,

$$\boxed{tf(t) \Leftrightarrow -\frac{d}{ds}F(s)} \tag{2.21}$$

Proof:

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st}dt$$

Differentiating with respect to s and applying *Leibnitz's rule** *for differentiation under the integral,* we obtain

$$\frac{d}{ds}F(s) = \frac{d}{ds}\int_0^\infty f(t)e^{-st}dt = \int_0^\infty \frac{\partial}{\partial s}e^{-st}f(t)dt = \int_0^\infty -te^{-st}f(t)dt = -\int_0^\infty [tf(t)]e^{-st}dt = -\mathcal{L}[tf(t)]$$

In general,

$$\boxed{t^n f(t) \Leftrightarrow (-1)^n \frac{d^n}{ds^n}F(s)} \tag{2.22}$$

The proof for $n \geq 2$ follows by taking the second and higher–order derivatives of $F(s)$ with respect to s.

2.2.7 Integration in Time Domain Property

This property states that *integration in time domain* corresponds to $F(s)$ divided by s plus the initial value of $f(t)$ at $t = 0^-$, also divided by s. That is,

---

* *This rule states that if a function of a parameter α is defined by the equation $F(\alpha) = \int_a^b f(x, \alpha)dx$ where f is some known function of integration x and the parameter α, a and b are constants independent of x and α, and the partial derivative $\partial f/\partial\alpha$ exists and it is continuous, then $\frac{dF}{d\alpha} = \int_a^b \frac{\partial(x, \alpha)}{\partial(\alpha)}dx$.*

$$\boxed{\int_{-\infty}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(s)}{s} + \frac{f(0^-)}{s}} \tag{2.23}$$

Proof:

We begin by expressing the integral on the left side of (2.23) as two integrals, that is,

$$\int_{-\infty}^{t} f(\tau)d\tau = \int_{-\infty}^{0} f(\tau)d\tau + \int_{0}^{t} f(\tau)d\tau \tag{2.24}$$

The first integral on the right side of (2.24), represents a constant value since neither the upper, nor the lower limits of integration are functions of time, and this constant is an initial condition denoted as $f(0^-)$. We will find the Laplace transform of this constant, the transform of the second integral on the right side of (2.24), and will prove (2.23) by the linearity property. Thus,

$$\begin{aligned} \mathcal{L}\{f(0^-)\} &= \int_0^{\infty} f(0^-)e^{-st}dt = f(0^-)\int_0^{\infty} e^{-st}dt = f(0^-)\frac{e^{-st}}{-s}\Bigg|_0^{\infty} \\ &= f(0^-)\times 0 - \left(-\frac{f(0^-)}{s}\right) = \frac{f(0^-)}{s} \end{aligned} \tag{2.25}$$

This is the value of the first integral in (2.24). Next, we will show that

$$\int_{0}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(s)}{s}$$

We let

$$g(t) = \int_{0}^{t} f(\tau)d\tau$$

then,

$$g'(t) = f(\tau)$$

and

$$g(0) = \int_{0}^{0} f(\tau)d\tau = 0$$

Now,

$$\mathcal{L}\{g'(t)\} = G(s) = s\mathcal{L}\{g(t)\} - g(0^-) = G(s) - 0$$

$$s\mathcal{L}\{g(t)\} = G(s)$$

$$\mathcal{L}\{g(t)\} = \frac{G(s)}{s}$$

$$\mathcal{L}\left\{\int_0^t f(\tau)d\tau\right\} = \frac{F(s)}{s} \tag{2.26}$$

and the proof of (2.23) follows from (2.25) and (2.26).

2.2.8 Integration in Complex Frequency Domain Property

This property states that *integration in complex frequency domain* with respect to s corresponds to division of a time function $f(t)$ by the variable t, provided that the limit $\lim_{t\to 0}\frac{f(t)}{t}$ exists. Thus,

$$\boxed{\frac{f(t)}{t} \Leftrightarrow \int_s^\infty F(s)ds} \tag{2.27}$$

Proof:

$$F(s) = \int_0^\infty f(t)e^{-st}dt$$

Integrating both sides from s to ∞, we obtain

$$\int_s^\infty F(s)ds = \int_s^\infty\left[\int_0^\infty f(t)e^{-st}dt\right]ds$$

Next, we interchange the order of integration, i.e.,

$$\int_s^\infty F(s)ds = \int_0^\infty\left[\int_s^\infty e^{-st}ds\right]f(t)dt$$

and performing the inner integration on the right side integral with respect to s, we obtain

$$\int_s^\infty F(s)ds = \int_0^\infty\left[-\frac{1}{t}e^{-st}\Big|_s^\infty\right]f(t)dt = \int_0^\infty\frac{f(t)}{t}e^{-st}dt = \mathcal{L}\left\{\frac{f(t)}{t}\right\}$$

2.2.9 Time Periodicity Property

The *time periodicity* property states that a periodic function of time with period T corresponds to the integral $\int_0^T f(t)e^{-st}dt$ divided by $(1-e^{-sT})$ in the complex frequency domain. Thus, if we let $f(t)$ be a periodic function with period T, that is, $f(t) = f(t+nT)$, for $n = 1, 2, 3, \ldots$ we obtain the transform pair

$$f(t+nT) \Leftrightarrow \frac{\int_0^T f(t)e^{-st}dt}{1-e^{-sT}} \tag{2.28}$$

Proof:

The Laplace transform of a periodic function can be expressed as

$$\mathcal{L}\{f(t)\} = \int_0^\infty f(t)e^{-st}dt = \int_0^T f(t)e^{-st}dt + \int_T^{2T} f(t)e^{-st}dt + \int_{2T}^{3T} f(t)e^{-st}dt + \ldots$$

In the first integral of the right side, we let $t = \tau$, in the second $t = \tau + T$, in the third $t = \tau + 2T$, and so on. The areas under each period of $f(t)$ are equal, and thus the upper and lower limits of integration are the same for each integral. Then,

$$\mathcal{L}\{f(t)\} = \int_0^T f(\tau)e^{-s\tau}d\tau + \int_0^T f(\tau+T)e^{-s(\tau+T)}d\tau + \int_0^T f(\tau+2T)e^{-s(\tau+2T)}d\tau + \ldots \tag{2.29}$$

Since the function is periodic, i.e., $f(\tau) = f(\tau+T) = f(\tau+2T) = \ldots = f(\tau+nT)$, we can write (2.29) as

$$\mathcal{L}\{f(\tau)\} = (1 + e^{-sT} + e^{-2sT} + \ldots)\int_0^T f(\tau)e^{-s\tau}d\tau \tag{2.30}$$

By application of the binomial theorem, that is,

$$1 + a + a^2 + a^3 + \ldots = \frac{1}{1-a} \tag{2.31}$$

we find that expression (2.30) reduces to

$$\mathcal{L}\{f(\tau)\} = \frac{\int_0^T f(\tau)e^{-s\tau}d\tau}{1-e^{-sT}}$$

2.2.10 Initial Value Theorem

The *initial value theorem* states that the initial value $f(0^-)$ of the time function $f(t)$ can be found from its Laplace transform multiplied by s and letting $s \to \infty$. That is,

$$\lim_{t\to 0} f(t) = \lim_{s\to\infty} sF(s) = f(0^-) \tag{2.32}$$

Proof:

From the time domain differentiation property,

$$\frac{d}{dt}f(t) \Leftrightarrow sF(s) - f(0^-)$$

or

$$\mathcal{L}\left\{\frac{d}{dt}f(t)\right\} = sF(s) - f(0^-) = \int_0^\infty \frac{d}{dt}f(t)e^{-st}dt$$

Taking the limit of both sides by letting $s \to \infty$, we obtain

$$\lim_{s\to\infty}[sF(s)-f(0^-)] = \lim_{s\to\infty}\left[\lim_{\substack{T\to\infty\\ \varepsilon\to 0}}\int_\varepsilon^T \frac{d}{dt}f(t)e^{-st}dt\right]$$

Interchanging the limiting process, we obtain

$$\lim_{s\to\infty}[sF(s)-f(0^-)] = \lim_{\substack{T\to\infty\\ \varepsilon\to 0}}\int_\varepsilon^T \frac{d}{dt}f(t)\left[\lim_{s\to\infty}e^{-st}\right]dt$$

and since

$$\lim_{s\to\infty}e^{-st} = 0$$

the above expression reduces to

$$\lim_{s\to\infty}[sF(s)-f(0^-)] = 0$$

or

$$\lim_{s\to\infty}sF(s) = f(0^-)$$

2.2.11 Final Value Theorem

The *final value theorem* states that the final value $f(\infty)$ of the time function $f(t)$ can be found from its Laplace transform multiplied by s, then, letting $s \to 0$. That is,

$$\boxed{\lim_{t\to\infty}f(t) = \lim_{s\to 0}sF(s) = f(\infty)} \tag{2.33}$$

Proof:

From the time domain differentiation property,

$$\frac{d}{dt}f(t) \Leftrightarrow sF(s) - f(0^-)$$

or

$$\mathscr{L}\left\{\frac{d}{dt}f(t)\right\} = sF(s) - f(0^-) = \int_0^\infty \frac{d}{dt}f(t)e^{-st}dt$$

Taking the limit of both sides by letting $s \to 0$, we obtain

$$\lim_{s\to 0}[sF(s)-f(0^-)] = \lim_{s\to 0}\left[\lim_{\substack{T\to\infty\\ \varepsilon\to 0}}\int_\varepsilon^T \frac{d}{dt}f(t)e^{-st}dt\right]$$

and by interchanging the limiting process, the expression above is written as

$$\lim_{s\to 0}[sF(s)-f(0^-)] = \lim_{\substack{T\to\infty\\ \varepsilon\to 0}}\int_\varepsilon^T \frac{d}{dt}f(t)\left[\lim_{s\to 0}e^{-st}\right]dt$$

Also, since

$$\lim_{s\to 0}e^{-st} = 1$$

it reduces to

$$\lim_{s\to 0}[sF(s)-f(0^-)] = \lim_{\substack{T\to\infty\\ \varepsilon\to 0}}\int_\varepsilon^T \frac{d}{dt}f(t)dt = \lim_{\substack{T\to\infty\\ \varepsilon\to 0}}\int_\varepsilon^T f(t) = \lim_{\substack{T\to\infty\\ \varepsilon\to 0}}[f(T)-f(\varepsilon)] = f(\infty)-f(0^-)$$

Therefore,

$$\lim_{s\to 0}sF(s) = f(\infty)$$

2.2.12 Convolution in Time Domain Property

*Convolution** *in the time domain* corresponds to multiplication in the complex frequency domain, that is,

$$f_1(t)*f_2(t) \Leftrightarrow F_1(s)F_2(s) \tag{2.34}$$

* *Convolution is the process of overlapping two time functions* $f_1(t)$ *and* $f_2(t)$. *The convolution integral indicates the amount of overlap of one function as it is shifted over another function The convolution of two time functions* $f_1(t)$ *and* $f_2(t)$ *is denoted as* $f_1(t)*f_2(t)$, *and by definition,* $f_1(t)*f_2(t) = \int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)d\tau$ *where* τ *is a dummy variable. Convolution is discussed in detail in Chapter 6.*

Proof:

$$\begin{aligned}\mathcal{L}\{f_1(t)*f_2(t)\} &= \mathcal{L}\left[\int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)d\tau\right] = \int_0^{\infty}\left[\int_0^{\infty} f_1(\tau)f_2(t-\tau)d\tau\right]e^{-st}dt \\ &= \int_0^{\infty} f_1(\tau)\left[\int_0^{\infty} f_2(t-\tau)e^{-st}dt\right]d\tau\end{aligned} \tag{2.35}$$

We let $t-\tau = \lambda$; then, $t = \lambda+\tau$, and $dt = d\lambda$. Then, by substitution into (2.35),

$$\begin{aligned}\mathcal{L}\{f_1(t)*f_2(t)\} &= \int_0^{\infty} f_1(\tau)\left[\int_0^{\infty} f_2(\lambda)e^{-s(\lambda+\tau)}d\lambda\right]d\tau = \int_0^{\infty} f_1(\tau)e^{-s\tau}d\tau\int_0^{\infty} f_2(\lambda)e^{-s\lambda}d\lambda \\ &= F_1(s)F_2(s)\end{aligned}$$

2.2.13 Convolution in Complex Frequency Domain Property

Convolution in the complex frequency domain divided by $1/2\pi j$, corresponds to multiplication in the time domain. That is,

$$f_1(t)f_2(t) \Leftrightarrow \frac{1}{2\pi j}F_1(s)*F_2(s) \tag{2.36}$$

Proof:

$$\mathcal{L}\{f_1(t)f_2(t)\} = \int_0^{\infty} f_1(t)f_2(t)e^{-st}dt \tag{2.37}$$

and recalling that the Inverse Laplace transform from (2.2) is

$$f_1(t) = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F_1(\mu)e^{\mu t}d\mu$$

by substitution into (2.37), we obtain

$$\mathcal{L}\{f_1(t)f_2(t)\} = \int_0^{\infty}\left[\frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F_1(\mu)e^{\mu t}d\mu\right]f_2(t)e^{-st}dt = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F_1(\mu)\left[\int_0^{\infty} f_2(t)e^{-(s-\mu)t}dt\right]d\mu$$

We observe that the bracketed integral is $F_2(s-\mu)$; therefore,

$$\mathcal{L}\{f_1(t)f_2(t)\} = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F_1(\mu)F_2(s-\mu)d\mu = \frac{1}{2\pi j}F_1(s)*F_2(s)$$

For easy reference, the Laplace transform pairs and theorems are summarized in Table 2.1.

TABLE 2.1 Summary of Laplace Transform Properties and Theorems

	Property/Theorem	*Time Domain*	*Complex Frequency Domain*
1	Linearity	$c_1 f_1(t) + c_2 f_2(t) + \ldots + c_n f_n(t)$	$c_1 F_1(s) + c_2 F_2(s) + \ldots + c_n F_n(s)$
2	Time Shifting	$f(t-a)u_0(t-a)$	$e^{-as}F(s)$
3	Frequency Shifting	$e^{-as}f(t)$	$F(s+a)$
4	Time Scaling	$f(at)$	$\frac{1}{a}F\left(\frac{s}{a}\right)$
5	Time Differentiation See also (2.18) through (2.20)	$\frac{d}{dt}f(t)$	$sF(s) - f(0^-)$
6	Frequency Differentiation See also (2.22)	$tf(t)$	$-\frac{d}{ds}F(s)$
7	Time Integration	$\int_{-\infty}^{t} f(\tau)d\tau$	$\frac{F(s)}{s} + \frac{f(0^-)}{s}$
8	Frequency Integration	$\frac{f(t)}{t}$	$\int_{s}^{\infty} F(s)ds$
9	Time Periodicity	$f(t+nT)$	$\frac{\int_0^T f(t)e^{-st}dt}{1-e^{-sT}}$
10	Initial Value Theorem	$\lim_{t \to 0} f(t)$	$\lim_{s \to \infty} sF(s) = f(0^-)$
11	Final Value Theorem	$\lim_{t \to \infty} f(t)$	$\lim_{s \to 0} sF(s) = f(\infty)$
12	Time Convolution	$f_1(t)*f_2(t)$	$F_1(s)F_2(s)$
13	Frequency Convolution	$f_1(t)f_2(t)$	$\frac{1}{2\pi j}F_1(s)*F_2(s)$

2.3 The Laplace Transform of Common Functions of Time

In this section, we will derive the Laplace transform of common functions of time. They are presented in Subsections 2.3.1 through 2.3.11 below.

2.3.1 The Laplace Transform of the Unit Step Function $u_0(t)$

We begin with the definition of the Laplace transform, that is,

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st}dt$$

or

$$\mathcal{L}\{u_0(t)\} = \int_0^\infty 1e^{-st}dt = \frac{-e^{st}}{s}\bigg|_0^\infty = 0-\left(-\frac{1}{s}\right) = \frac{1}{s}$$

Thus, we have obtained the transform pair

$$\boxed{u_0(t) \Leftrightarrow \frac{1}{s}} \tag{2.38}$$

for $\mathrm{Re}\{s\} = \sigma > 0$.*

2.3.2 The Laplace Transform of the Ramp Function $u_1(t)$

We apply the definition

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st}dt$$

or

$$\mathcal{L}\{u_1(t)\} = \mathcal{L}\{t\} = \int_0^\infty te^{-st}dt$$

We will perform integration by parts by recalling that

$$\int udv = uv - \int vdu \tag{2.39}$$

We let

$$u = t \text{ and } dv = e^{-st}$$

then,

$$du = 1 \text{ and } v = \frac{-e^{-st}}{s}$$

* *This condition was established in relation (2.9), Page 2–2.*

By substitution into (2.39),

$$\mathcal{L}\{t\} = \frac{-te^{-st}}{s}\bigg|_0^\infty - \int_0^\infty \frac{-e^{-st}}{s}dt = \left[\frac{-te^{-st}}{s} - \frac{e^{-st}}{s^2}\right]\bigg|_0^\infty \tag{2.40}$$

Since the upper limit of integration in (2.40) produces an indeterminate form, we apply *L' Hôpital's rule**, that is,

$$\lim_{t\to\infty} te^{-st} = \lim_{t\to\infty} \frac{t}{e^{st}} = \lim_{t\to\infty} \frac{\frac{d}{dt}(t)}{\frac{d}{dt}(e^{st})} = \lim_{t\to\infty} \frac{1}{se^{st}} = 0$$

Evaluating the second term of (2.40), we obtain $\mathcal{L}\{t\} = \frac{1}{s^2}$

Thus, we have obtained the transform pair

$$\boxed{t \Leftrightarrow \frac{1}{s^2}} \tag{2.41}$$

for $\sigma > 0$.

2.3.3 The Laplace Transform of $t^n u_0(t)$

Before deriving the Laplace transform of this function, we digress to review the *gamma* or *generalized factorial function* $\Gamma(n)$ which is an *improper integral*† but converges (approaches a limit) for all $n > 0$. It is defined as

---

* *Often, the ratio of two functions, such as* $\frac{f(x)}{g(x)}$, *for some value of x, say a, results in an indeterminate form. To work around this problem, we consider the limit* $\lim_{x\to a}\frac{f(x)}{g(x)}$, *and we wish to find this limit, if it exists. To find this limit, we use L'Hôpital's rule which states that if* $f(a) = g(a) = 0$, *and if the limit* $\frac{d}{dx}f(x)/\frac{d}{dx}g(x)$ *as x approaches a exists, then,*

$$\lim_{x\to a}\frac{f(x)}{g(x)} = \lim_{x\to a}\left(\frac{d}{dx}f(x)/\frac{d}{dx}g(x)\right)$$

† *Improper integrals are two types and these are:*

a. $\int_a^b f(x)dx$ *where the limits of integration a or b or both are infinite*

b. $\int_a^b f(x)dx$ *where f(x) becomes infinite at a value x between the lower and upper limits of integration inclusive.*

$$\boxed{\Gamma(n) = \int_0^\infty x^{n-1}e^{-x}dx} \tag{2.42}$$

We will now derive the basic properties of the gamma function, and its relation to the well known factorial function

$$n! = n(n-1)(n-2)\cdot\ \cdot\ 3\cdot 2\cdot 1$$

The integral of (2.42) can be evaluated by performing integration by parts. Thus, in (2.42) we let

$$u = e^{-x} \quad \text{and} \quad dv = x^{n-1}$$

Then,

$$du = -e^{-x}dx \quad \text{and} \quad v = \frac{x^n}{n}$$

and (2.42) is written as

$$\Gamma(n) = \frac{x^n e^{-x}}{n}\bigg|_{x=0}^{\infty} + \frac{1}{n}\int_0^\infty x^n e^{-x}dx \tag{2.43}$$

With the condition that $n > 0$, the first term on the right side of (2.43) vanishes at the lower limit $x = 0$. It also vanishes at the upper limit as $x \to \infty$. This can be proved with L' Hôpital's rule by differentiating both numerator and denominator m times, where $m \geq n$. Then,

$$\lim_{x\to\infty}\frac{x^n e^{-x}}{n} = \lim_{x\to\infty}\frac{x^n}{ne^x} = \lim_{x\to\infty}\frac{\dfrac{d^m}{dx^m}x^n}{\dfrac{d^m}{dx^m}ne^x} = \lim_{x\to\infty}\frac{\dfrac{d^{m-1}}{dx^{m-1}}nx^{n-1}}{\dfrac{d^{m-1}}{dx^{m-1}}ne^x} = \ldots$$

$$= \lim_{x\to\infty}\frac{n(n-1)(n-2)\ldots(n-m+1)x^{n-m}}{ne^x} = \lim_{x\to\infty}\frac{(n-1)(n-2)\ldots(n-m+1)}{x^{m-n}e^x} = 0$$

Therefore, (2.43) reduces to

$$\Gamma(n) = \frac{1}{n}\int_0^\infty x^n e^{-x}dx$$

and with (2.42), we have

$$\Gamma(n) = \int_0^\infty x^{n-1}e^{-x}dx = \frac{1}{n}\int_0^\infty x^n e^{-x}dx \tag{2.44}$$

By comparing the integrals in (2.44), we observe that

$$\boxed{\Gamma(n) = \frac{\Gamma(n+1)}{n}} \tag{2.45}$$

or

$$\boxed{n\Gamma(n) = \Gamma(n+1)} \tag{2.46}$$

It is convenient to use (2.45) for $n<0$, and (2.46) for $n>0$. From (2.45), we see that $\Gamma(n)$ becomes infinite as $n \to 0$.

For $n = 1$, (2.42) yields

$$\Gamma(1) = \int_0^\infty e^{-x}dx = -e^{-x}\Big|_0^\infty = 1 \tag{2.47}$$

and thus we have obtained the important relation,

$$\Gamma(1) = 1 \tag{2.48}$$

From the recurring relation of (2.46), we obtain

$$\begin{aligned}\Gamma(2) &= 1 \cdot \Gamma(1) = 1\\ \Gamma(3) &= 2 \cdot \Gamma(2) = 2 \cdot 1 = 2!\\ \Gamma(4) &= 3 \cdot \Gamma(3) = 3 \cdot 2 = 3!\end{aligned} \tag{2.49}$$

and in general

$$\boxed{\Gamma(n+1) = n!} \tag{2.50}$$

for $n = 1, 2, 3, \ldots$

The formula of (2.50) is a noteworthy relation; it establishes the relationship between the $\Gamma(n)$ function and the factorial $n!$

We now return to the problem of finding the Laplace transform pair for $t^n u_0 t$, that is,

$$\mathscr{L}\{t^n u_0 t\} = \int_0^\infty t^n e^{-st}dt \tag{2.51}$$

To make this integral resemble the integral of the gamma function, we let $st = y$, or $t = y/s$, and thus $dt = dy/s$. Now, we rewrite (2.51) as

$$\mathscr{L}\{t^n u_0 t\} = \int_0^\infty \left(\frac{y}{s}\right)^n e^{-y} d\left(\frac{y}{s}\right) = \frac{1}{s^{n+1}}\int_0^\infty y^n e^{-y}dy = \frac{\Gamma(n+1)}{s^{n+1}} = \frac{n!}{s^{n+1}}$$

Therefore, we have obtained the transform pair

$$\boxed{t^n u_0(t) \Leftrightarrow \frac{n!}{s^{n+1}}} \tag{2.52}$$

for positive integers of n and $\sigma > 0$.

2.3.4 The Laplace Transform of the Delta Function $\delta(t)$

We apply the definition

$$\mathcal{L}\{\delta(t)\} = \int_0^\infty \delta(t)e^{-st}dt$$

and using the sifting property of the delta function,* we obtain

$$\mathcal{L}\{\delta(t)\} = \int_0^\infty \delta(t)e^{-st}dt = e^{-s(0)} = 1$$

Thus, we have the transform pair

$$\boxed{\delta(t) \Leftrightarrow 1} \tag{2.53}$$

for all σ.

2.3.5 The Laplace Transform of the Delayed Delta Function $\delta(t-a)$

We apply the definition

$$\mathcal{L}\{\delta(t-a)\} = \int_0^\infty \delta(t-a)e^{-st}dt$$

and again, using the sifting property of the delta function, we obtain

$$\mathcal{L}\{\delta(t-a)\} = \int_0^\infty \delta(t-a)e^{-st}dt = e^{-as}$$

Thus, we have the transform pair

$$\boxed{\delta(t-a) \Leftrightarrow e^{-as}} \tag{2.54}$$

for $\sigma > 0$.

* *The sifting property of the* $\delta(t)$ *is described in Subsection 1.4.2, Chapter 1, Page 1–13.*

2.3.6 The Laplace Transform of $e^{-at}u_0(t)$

We apply the definition

$$\mathcal{L}\{e^{-at}u_0(t)\} = \int_0^\infty e^{-at}e^{-st}dt = \int_0^\infty e^{-(s+a)t}dt = \left(-\frac{1}{s+a}\right)e^{-(s+a)t}\Big|_0^\infty = \frac{1}{s+a}$$

Thus, we have the transform pair

$$\boxed{e^{-at}u_0(t) \Leftrightarrow \frac{1}{s+a}} \tag{2.55}$$

for $\sigma > -a$.

2.3.7 The Laplace Transform of $t^n e^{-at}u_0(t)$

For this derivation, we will use the transform pair of (2.52), i.e.,

$$t^n u_0(t) \Leftrightarrow \frac{n!}{s^{n+1}} \tag{2.56}$$

and the frequency shifting property of (2.14), that is,

$$e^{-at}f(t) \Leftrightarrow F(s+a) \tag{2.57}$$

Then, replacing s with $s+a$ in (2.56), we obtain the transform pair

$$\boxed{t^n e^{-at}u_0(t) \Leftrightarrow \frac{n!}{(s+a)^{n+1}}} \tag{2.58}$$

where n is a positive integer, and $\sigma > -a$. Thus, for $n = 1$, we obtain the transform pair

$$\boxed{te^{-at}u_0(t) \Leftrightarrow \frac{1}{(s+a)^2}} \tag{2.59}$$

for $\sigma > -a$.

For $n = 2$, we obtain the transform

$$\boxed{t^2 e^{-at}u_0(t) \Leftrightarrow \frac{2!}{(s+a)^3}} \tag{2.60}$$

and in general,

$$\boxed{t^n e^{-at} u_0(t) \Leftrightarrow \frac{n!}{(s+a)^{n+1}}} \tag{2.61}$$

for $\sigma > -a$.

2.3.8 The Laplace Transform of $\sin\omega t\, u_0(t)$

We apply the definition

$$\mathcal{L}\{\sin\omega t\, u_0(t)\} = \int_0^\infty (\sin\omega t)e^{-st}dt = \lim_{a\to\infty}\int_0^a (\sin\omega t)e^{-st}dt$$

and from tables of integrals*

$$\int e^{ax}\sin bx\,dx = \frac{e^{ax}(a\sin bx - b\cos bx)}{a^2+b^2}$$

Then,

$$\mathcal{L}\{\sin\omega t\, u_0(t)\} = \lim_{a\to\infty}\frac{e^{-st}(-s\sin\omega t - \omega\cos\omega t)}{s^2+\omega^2}\Bigg|_0^a$$

$$= \lim_{a\to\infty}\left[\frac{e^{-as}(-s\sin\omega a - \omega\cos\omega a)}{s^2+\omega^2} + \frac{\omega}{s^2+\omega^2}\right] = \frac{\omega}{s^2+\omega^2}$$

Thus, we have obtained the transform pair

$$\boxed{\sin\omega t\, u_0 t \Leftrightarrow \frac{\omega}{s^2+\omega^2}} \tag{2.62}$$

for $\sigma > 0$.

2.3.9 The Laplace Transform of $\cos\omega t\, u_0(t)$

We apply the definition

$$\mathcal{L}\{\cos\omega t\, u_0(t)\} = \int_0^\infty (\cos\omega t)e^{-st}dt = \lim_{a\to\infty}\int_0^a (\cos\omega t)e^{-st}dt$$

---

* *This can also be derived from* $\sin\omega t = \frac{1}{j2}(e^{j\omega t} - e^{-j\omega t})$, *and the use of (2.55) where* $e^{-at}u_0(t) \Leftrightarrow \frac{1}{s+a}$. *By the linearity property, the sum of these terms corresponds to the sum of their Laplace transforms. Therefore,* $\mathcal{L}[\sin\omega t u_0(t)] = \frac{1}{j2}\left(\frac{1}{s-j\omega} - \frac{1}{s+j\omega}\right) = \frac{\omega}{s^2+\omega^2}$

and from tables of integrals*

$$\int e^{ax}\cos bx\,dx = \frac{e^{ax}(a\cos bx + b\sin bx)}{a^2 + b^2}$$

Then,

$$\mathcal{L}\{\cos\omega t\ u_0(t)\} = \lim_{a\to\infty}\frac{e^{-st}(-s\cos\omega t + \omega\sin\omega t)}{s^2+\omega^2}\Bigg|_0^a$$

$$= \lim_{a\to\infty}\left[\frac{e^{-as}(-s\cos\omega a + \omega\sin\omega a)}{s^2+\omega^2} + \frac{s}{s^2+\omega^2}\right] = \frac{s}{s^2+\omega^2}$$

Thus, we have the fransform pair

$$\boxed{\cos\omega t\ u_0 t \Leftrightarrow \frac{s}{s^2+\omega^2}} \tag{2.63}$$

for $\sigma > 0$.

2.3.10 The Laplace Transform of $e^{-at}\sin\omega t\ u_0(t)$

From (2.62),

$$\sin\omega t u_0 t \Leftrightarrow \frac{\omega}{s^2+\omega^2}$$

Using the frequency shifting property of (2.14), that is,

$$e^{-at}f(t) \Leftrightarrow F(s+a) \tag{2.64}$$

we replace s with $s + a$, and we obtain

$$\boxed{e^{-at}\sin\omega t\ u_0(t) \Leftrightarrow \frac{\omega}{(s+a)^2+\omega^2}} \tag{2.65}$$

for $\sigma > 0$ and $a > 0$.

---

* *We can use the relation* $\cos\omega t = \frac{1}{2}(e^{j\omega t} + e^{-j\omega t})$ *and the linearity property, as in the derivation of the transform of* $\sin\omega t$ *on the footnote of the previous page. We can also use the transform pair* $\frac{d}{dt}f(t) \Leftrightarrow sF(s) - f(0^-)$; *this is the time differentiation property of (2.16). Applying this transform pair for this derivation, we obtain* $\mathcal{L}[\cos\omega t u_0(t)] = L\left[\frac{1}{\omega}\frac{d}{dt}\sin\omega t u_0(t)\right] = \frac{1}{\omega}\mathcal{L}\left[\frac{d}{dt}\sin\omega t u_0(t)\right] = \frac{1}{\omega}s\frac{\omega}{s^2+\omega^2} = \frac{s}{s^2+\omega^2}$

2.3.11 The Laplace Transform of $e^{-at}\cos\omega t\, u_0(t)$

From (2.63),

$$\cos\omega t\, u_0(t) \Leftrightarrow \frac{s}{s^2+\omega^2}$$

and using the frequency shifting property of (2.14), we replace s with $s+a$, and we obtain

$$\boxed{e^{-at}\cos\omega t\, u_0(t) \Leftrightarrow \frac{s+a}{(s+a)^2+\omega^2}} \tag{2.66}$$

for $\sigma > 0$ and $a > 0$.

For easy reference, we have summarized the above derivations in Table 2.2.

TABLE 2.2 Laplace Transform Pairs for Common Functions

	$\mathbf{f(t)}$	$\mathbf{F(s)}$
1	$u_0(t)$	$1/s$
2	$tu_0(t)$	$1/s^2$
3	$t^n u_0(t)$	$\frac{n!}{s^{n+1}}$
4	$\delta(t)$	1
5	$\delta(t-a)$	e^{-as}
6	$e^{-at}u_0(t)$	$\frac{1}{s+a}$
7	$t^n e^{-at}u_0(t)$	$\frac{n!}{(s+a)^{n+1}}$
8	$\sin\omega t\, u_0(t)$	$\frac{\omega}{s^2+\omega^2}$
9	$\cos\omega t\, u_0(t)$	$\frac{s}{s^2+\omega^2}$
10	$e^{-at}\sin\omega t\, u_0(t)$	$\frac{\omega}{(s+a)^2+\omega^2}$
11	$e^{-at}\cos\omega t\, u_0(t)$	$\frac{s+a}{(s+a)^2+\omega^2}$

2.4 The Laplace Transform of Common Waveforms

In this section, we will present procedures for deriving the Laplace transform of common waveforms using the transform pairs of Tables 1 and 2. The derivations are described in Subsections 2.4.1 through 2.4.5 below.

2.4.1 The Laplace Transform of a Pulse

The waveform of a pulse, denoted as $f_P(t)$, is shown in Figure 2.1.

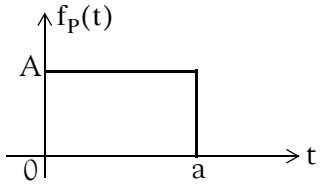


Figure 2.1. Waveform for a pulse

We first express the given waveform as a sum of unit step functions as we've learned in Chapter 1. Then,

$$f_P(t) = A[u_0(t) - u_0(t-a)] \tag{2.67}$$

From Table 2.1, Page 2–13,

$$f(t-a)u_0(t-a) \Leftrightarrow e^{-as}F(s)$$

and from Table 2.2, Page 2–22

$$u_0(t) \Leftrightarrow 1/s$$

Thus,

$$Au_0(t) \Leftrightarrow A/s$$

and

$$Au_0(t-a) \Leftrightarrow e^{-as}\frac{A}{s}$$

Then, in accordance with the linearity property, the Laplace transform of the pulse of Figure 2.1 is

$$\boxed{A[u_0(t) - u_0(t-a)] \Leftrightarrow \frac{A}{s} - e^{-as}\frac{A}{s} = \frac{A}{s}(1 - e^{-as})}$$

2.4.2 The Laplace Transform of a Linear Segment

The waveform of a linear segment, denoted as $f_L(t)$, is shown in Figure 2.2.

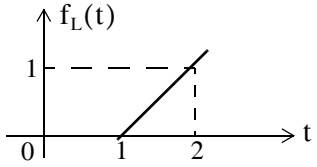


Figure 2.2. Waveform for a linear segment

We must first derive the equation of the linear segment. This is shown in Figure 2.3.

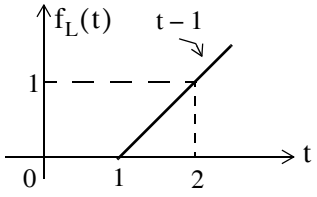


Figure 2.3. Waveform for a linear segment with the equation that describes it

Next, we express the given waveform in terms of the unit step function as follows:

$$f_L(t) = (t-1)u_0(t-1)$$

From Table 2.1, Page 2–13,

$$f(t-a)u_0(t-a) \Leftrightarrow e^{-as}F(s)$$

and from Table 2.2, Page 2–22,

$$tu_0(t) \Leftrightarrow \frac{1}{s^2}$$

Therefore, the Laplace transform of the linear segment of Figure 2.2 is

$$\boxed{(t-1)u_0(t-1) \Leftrightarrow e^{-s}\frac{1}{s^2}} \qquad (2.68)$$

2.4.3 The Laplace Transform of a Triangular Waveform

The waveform of a triangular waveform, denoted as $f_T(t)$, is shown in Figure 2.4.

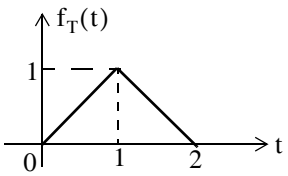


Figure 2.4. Triangular waveform

The equations of the linear segments are shown in Figure 2.5.

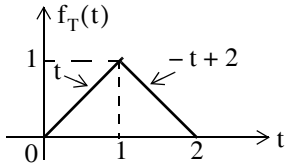


Figure 2.5. Triangular waveform with the equations of the linear segments

Next, we express the given waveform in terms of the unit step function.

$$\begin{aligned} f_T(t) &= t[u_0(t)-u_0(t-1)]+(-t+2)[u_0(t-1)-u_0(t-2)] \\ &= tu_0(t)-tu_0(t-1)-tu_0(t-1)+2u_0(t-1)+tu_0(t-2)-2u_0(t-2) \end{aligned}$$

Collecting like terms, we obtain

$$f_T(t) = tu_0(t)-2(t-1)u_0(t-1)+(t-2)u_0(t-2)$$

From Table 2.1, Page 2–13,

$$f(t-a)u_0(t-a) \Leftrightarrow e^{-as}F(s)$$

and from Table 2.2, Page 2–22,

$$tu_0(t) \Leftrightarrow \frac{1}{s^2}$$

Then,

$$tu_0(t)-2(t-1)u_0(t-1)+(t-2)u_0(t-2) \Leftrightarrow \frac{1}{s^2}-2e^{-s}\frac{1}{s^2}+e^{-2s}\frac{1}{s^2}$$

or

$$tu_0(t)-2(t-1)u_0(t-1)+(t-2)u_0(t-2) \Leftrightarrow \frac{1}{s^2}(1-2e^{-s}+e^{-2s})$$

Therefore, the Laplace transform of the triangular waveform of Figure 2.4 is

$$\boxed{f_T(t) \Leftrightarrow \frac{1}{s^2}(1-e^{-s})^2} \tag{2.69}$$

2.4.4 The Laplace Transform of a Rectangular Periodic Waveform

The waveform of a rectangular periodic waveform, denoted as $f_R(t)$, is shown in Figure 2.6. This is a periodic waveform with period $T = 2a$, and we can apply the time periodicity property

$$\mathcal{L}\{f(\tau)\} = \frac{\int_0^T f(\tau)e^{-s\tau}d\tau}{1-e^{-sT}}$$

where the denominator represents the periodicity of f(t).

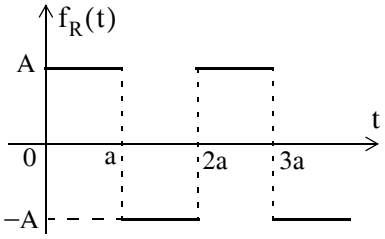


Figure 2.6. Rectangular periodic waveform

For this waveform,

$$\mathcal{L}\{f_R(t)\} = \frac{1}{1-e^{-2as}}\int_0^{2a} f_R(t)e^{-st}dt = \frac{1}{1-e^{-2as}}\left[\int_0^{a} Ae^{-st}dt + \int_a^{2a}(-A)e^{-st}dt\right]$$

$$= \frac{A}{1-e^{-2as}}\left[\frac{-e^{-st}}{s}\bigg|_0^a + \frac{e^{-st}}{s}\bigg|_a^{2a}\right]$$

$$\mathcal{L}\{f_R(t)\} = \frac{A}{s(1-e^{-2as})}(-e^{-as}+1+e^{-2as}-e^{-as})$$

$$= \frac{A}{s(1-e^{-2as})}(1-2e^{-as}+e^{-2as}) = \frac{A(1-e^{-as})^2}{s(1+e^{-as})(1-e^{-as})}$$

$$= \frac{A}{s}\frac{(1-e^{-as})}{(1+e^{-as})} = \frac{A}{s}\left(\frac{e^{as/2}e^{-as/2}-e^{-as/2}e^{-as/2}}{e^{as/2}e^{-as/2}+e^{-as/2}e^{-as/2}}\right)$$

$$= \frac{A}{s}\frac{e^{-as/2}}{e^{-as/2}}\left(\frac{e^{as/2}-e^{-as/2}}{e^{as/2}+e^{-as/2}}\right) = \frac{A}{s}\frac{\sinh(as/2)}{\cosh(as/2)}$$

$$\boxed{f_R(t) \Leftrightarrow \frac{A}{s}\tanh\left(\frac{as}{2}\right)} \tag{2.70}$$

2.4.5 The Laplace Transform of a Half–Rectified Sine Waveform

The waveform of a half-rectified sine waveform, denoted as $f_{HW}(t)$, is shown in Figure 2.7. This is a periodic waveform with period $T = 2a$, and we can apply the time periodicity property

$$\mathcal{L}\{f(\tau)\} = \frac{\int_0^T f(\tau)e^{-s\tau}d\tau}{1-e^{-sT}}$$

where the denominator represents the periodicity of f(t).

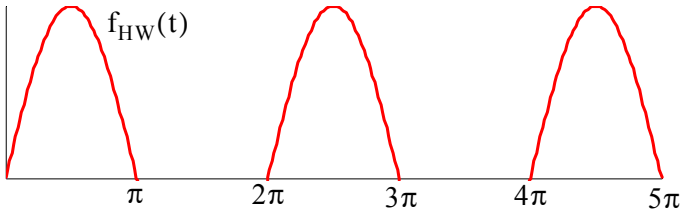


Figure 2.7. Half-rectified sine waveform*

For this waveform,

$$\mathcal{L}\{f_{HW}(t)\} = \frac{1}{1-e^{-2\pi s}}\int_0^{2\pi} f(t)e^{-st}dt = \frac{1}{1-e^{-2\pi s}}\int_0^{\pi} \sin t e^{-st}dt$$

$$= \frac{1}{1-e^{-2\pi s}}\left[\frac{e^{-st}(s\sin t - \cos t)}{s^2+1}\right]\Bigg|_0^{\pi} = \frac{1}{(s^2+1)}\frac{(1+e^{-\pi s})}{(1-e^{-2\pi s})}$$

$$\mathcal{L}\{f_{HW}(t)\} = \frac{1}{(s^2+1)}\frac{(1+e^{-\pi s})}{(1+e^{-\pi s})(1-e^{-\pi s})}$$

$$\boxed{f_{HW}(t) \Leftrightarrow \frac{1}{(s^2+1)(1-e^{-\pi s})}} \qquad (2.71)$$

2.5 Using MATLAB for Finding the Laplace Transforms of Time Functions

We can use the MATLAB function **laplace** to find the Laplace transform of a time function. For examples, please type

```
help laplace
```

in MATLAB's Command prompt.

We will be using this function extensively in the subsequent chapters of this book.

* *This waveform was produced with the following MATLAB script:*

```
t=0:pi/64:5*pi; x=sin(t); y=sin(t−2*pi); z=sin(t−4*pi); plot(t,x,t,y,t,z); axis([0 5*pi 0 1])
```

2.6 Summary

- The two–sided or bilateral Laplace Transform pair is defined as

$$\mathcal{L}\{f(t)\} = F(s) = \int_{-\infty}^{\infty} f(t)e^{-st}dt$$

$$\mathcal{L}^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F(s)e^{st}ds$$

  where $\mathcal{L}\{f(t)\}$ denotes the Laplace transform of the time function $f(t)$, $\mathcal{L}^{-1}\{F(s)\}$ denotes the Inverse Laplace transform, and s is a complex variable whose real part is σ, and imaginary part ω, that is, $s = \sigma + j\omega$.

- The unilateral or one–sided Laplace transform defined as

$$\mathcal{L}\{f(t)\} = F(s) = \int_{t_0}^{\infty} f(t)e^{-st}dt = \int_{0}^{\infty} f(t)e^{-st}dt$$

- We denote transformation from the time domain to the complex frequency domain, and vice versa, as

$$f(t) \Leftrightarrow F(s)$$

- The linearity property states that

$$c_1 f_1(t) + c_2 f_2(t) + \ldots + c_n f_n(t) \Leftrightarrow c_1 F_1(s) + c_2 F_2(s) + \ldots + c_n F_n(s)$$

- The time shifting property states that

$$f(t-a)u_0(t-a) \Leftrightarrow e^{-as}F(s)$$

- The frequency shifting property states that

$$e^{-at}f(t) \Leftrightarrow F(s+a)$$

- The scaling property states that

$$f(at) \Leftrightarrow \frac{1}{a}F\left(\frac{s}{a}\right)$$

- The differentiation in time domain property states that

$$f'(t) = \frac{d}{dt}f(t) \Leftrightarrow sF(s) - f(0^-)$$

  Also,

$$\frac{d^2}{dt^2} f(t) \Leftrightarrow s^2F(s) - sf(0^-) - f'(0^-)$$

$$\frac{d^3}{dt^3} f(t) \Leftrightarrow s^3F(s) - s^2f(0^-) - sf'(0^-) - f''(0^-)$$

and in general

$$\frac{d^n}{dt^n} f(t) \Leftrightarrow s^nF(s) - s^{n-1}f(0^-) - s^{n-2}f'(0^-) - \ldots - f^{n-1}(0^-)$$

where the terms $f(0^-)$, $f'(0^-)$, $f''(0^-)$, and so on, represent the initial conditions.

- The differentiation in complex frequency domain property states that

$$tf(t) \Leftrightarrow -\frac{d}{ds}F(s)$$

and in general,

$$t^nf(t) \Leftrightarrow (-1)^n\frac{d^n}{ds^n}F(s)$$

- The integration in time domain property states that

$$\int_{-\infty}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(s)}{s} + \frac{f(0^-)}{s}$$

- The integration in complex frequency domain property states that

$$\frac{f(t)}{t} \Leftrightarrow \int_s^{\infty} F(s)ds$$

provided that the limit $\lim_{t \to 0} \frac{f(t)}{t}$ exists.

- The time periodicity property states that

$$f(t + nT) \Leftrightarrow \frac{\int_0^T f(t)e^{-st}dt}{1 - e^{-sT}}$$

- The initial value theorem states that

$$\lim_{t \to 0} f(t) = \lim_{s \to \infty} sF(s) = f(0^-)$$

- The final value theorem states that

$$\lim_{t \to \infty} f(t) = \lim_{s \to 0} sF(s) = f(\infty)$$

- Convolution in the time domain corresponds to multiplication in the complex frequency domain, that is,

$$f_1(t)*f_2(t) \Leftrightarrow F_1(s)F_2(s)$$

- Convolution in the complex frequency domain divided by $1/2\pi j$, corresponds to multiplication in the time domain. That is,

$$f_1(t)f_2(t) \Leftrightarrow \frac{1}{2\pi j} F_1(s)*F_2(s)$$

- The Laplace transforms of some common functions of time are shown in Table 2.1, Page 2–13
- The Laplace transforms of some common waveforms are shown in Table 2.2, Page2–22
- We can use the MATLAB function **laplace** to find the Laplace transform of a time function

2.7 Exercises

1. Derive the Laplace transform of the following time domain functions:

a. 12 b. $6u_0(t)$ c. $24u_0(t-12)$ d. $5tu_0(t)$ e. $4t^5u_0(t)$

2. Derive the Laplace transform of the following time domain functions:

a. $j8$ b. $j5\angle -90°$ c. $5e^{-5t}u_0(t)$ d. $8t^7e^{-5t}u_0(t)$ e. $15\delta(t-4)$

3. Derive the Laplace transform of the following time domain functions:

a. $(t^3+3t^2+4t+3)u_0(t)$ b. $3(2t-3)\delta(t-3)$

c. $(3\sin 5t)u_0(t)$ d. $(5\cos 3t)u_0(t)$

e. $(2\tan 4t)u_0(t)$ Be careful with this! Comment and you may skip derivation.

4. Derive the Laplace transform of the following time domain functions:

a. $3t(\sin 5t)u_0(t)$ b. $2t^2(\cos 3t)u_0(t)$ c. $2e^{-5t}\sin 5t$

d. $8e^{-3t}\cos 4t$ e. $(\cos t)\delta(t-\pi/4)$

5. Derive the Laplace transform of the following time domain functions:

a. $5tu_0(t-3)$ b. $(2t^2-5t+4)u_0(t-3)$ c. $(t-3)e^{-2t}u_0(t-2)$

d. $(2t-4)e^{2(t-2)}u_0(t-3)$ e. $4te^{-3t}(\cos 2t)u_0(t)$

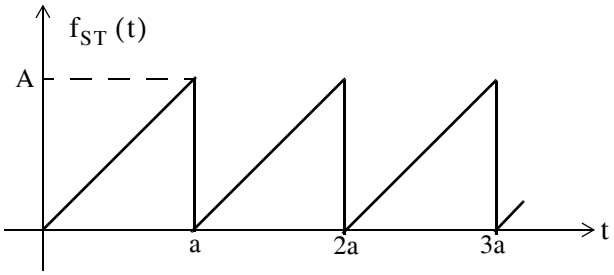
6. Derive the Laplace transform of the following time domain functions:

a. $\frac{d}{dt}(\sin 3t)$ b. $\frac{d}{dt}(3e^{-4t})$ c. $\frac{d}{dt}(t^2\cos 2t)$ d. $\frac{d}{dt}(e^{-2t}\sin 2t)$ e. $\frac{d}{dt}(t^2e^{-2t})$

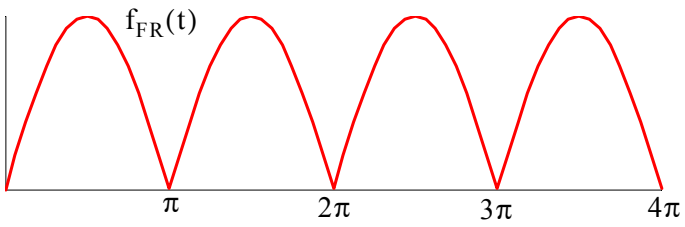
7. Derive the Laplace transform of the following time domain functions:

a. $\frac{\sin t}{t}$ b. $\int_0^t \frac{\sin\tau}{\tau}d\tau$ c. $\frac{\sin at}{t}$ d. $\int_t^\infty \frac{\cos\tau}{\tau}d\tau$ e. $\int_t^\infty \frac{e^{-\tau}}{\tau}d\tau$

8. Derive the Laplace transform for the sawtooth waveform $f_{ST}(t)$ below.



9. Derive the Laplace transform for the full–rectified waveform $f_{FR}(t)$ below.



Write a simple MATLAB script that will produce the waveform above.

2.8 Solutions to End–of–Chapter Exercises

1. From the definition of the Laplace transform or from Table 2.2, Page 2–22, we obtain:

a $12/s$ b. $6/s$ c. $e^{-12s}\cdot\frac{24}{s}$ d. $5/s^2$ e. $4\cdot\frac{5!}{s^6}$

2. From the definition of the Laplace transform or from Table 2.2, Page 2–22, we obtain:

a. $j8/s$ b. $5/s$ c. $\frac{5}{s+5}$ d. $8\cdot\frac{7!}{(s+5)^8}$ e. $15e^{-4s}$

3.

a. From Table 2.2, Page 2–22, and the linearity property, we obtain $\frac{3!}{s^4}+\frac{3\times 2!}{s^3}+\frac{4}{s^2}+\frac{3}{s}$

b. $3(2t-3)\delta(t-3) = 3(2t-3)\big|_{t=3}\delta(t-3) = 9\delta(t-3)$ and $9\delta(t-3)\Leftrightarrow 9e^{-3s}$

c. $3\cdot\frac{5}{s^2+5^2}$ d. $5\cdot\frac{s}{s^2+3^2}$ e. $2\tan 4t = 2\cdot\frac{\sin 4t}{\cos 4t}\Leftrightarrow 2\cdot\frac{4/(s^2+2^2)}{s/(s^2+2^2)}=\frac{8}{s}$

This answer for part (e) looks suspicious because $8/s\Leftrightarrow 8u_0(t)$ and the Laplace transform is unilateral, that is, there is one–to–one correspondence between the time domain and the complex frequency domain. The fallacy with this procedure is that we assumed that if $f_1(t)\Leftrightarrow F_1(s)$ and $f_2(t)\Leftrightarrow F_2(s)$, we cannot conclude that $\frac{f_1(t)}{f_2(t)}\Leftrightarrow\frac{F_1(s)}{F_2(s)}$. For this exercise $f_1(t)\cdot f_2(t) = \sin 4t\cdot\frac{1}{\cos 4t}$, and as we've learned, multiplication in the time domain corresponds to convolution in the complex frequency domain. Accordingly, we must use the Laplace transform definition $\int_0^\infty(2\tan 4t)e^{-st}dt$ and this requires integration by parts. We skip this analytical derivation. The interested reader may try to find the answer with the MATLAB script

```
syms s t; 2*laplace(sin(4*t)/cos(4*t))
```

4. From (2.22), Page 2–6,

$$t^n f(t)\Leftrightarrow(-1)^n\frac{d^n}{ds^n}F(s)$$

a.

$$3(-1)^1\frac{d}{ds}\left(\frac{5}{s^2+5^2}\right) = -3\left[\frac{-5\cdot(2s)}{(s^2+25)^2}\right] = \frac{30s}{(s^2+25)^2}$$

b.

$$2(-1)^2\frac{d^2}{ds^2}\left(\frac{s}{s^2+3^2}\right) = 2\frac{d}{ds}\left[\frac{s^2+3^2-s(2s)}{(s^2+9)^2}\right] = 2\frac{d}{ds}\left(\frac{-s^2+9}{(s^2+9)^2}\right)$$

$$= 2\cdot\frac{(s^2+9)^2(-2s)-2(s^2+9)(2s)(-s^2+9)}{(s^2+9)^4}$$

$$= 2\cdot\frac{(s^2+9)(-2s)-4s(-s^2+9)}{(s^2+9)^3} = 2\cdot\frac{-2s^3-18s+4s^3-36s}{(s^2+9)^3}$$

$$= 2\cdot\frac{2s^3-54s}{(s^2+9)^3} = 2\cdot\frac{2s(s^2-27)}{(s^2+9)^3} = \frac{4s(s^2-27)}{(s^2+9)^3}$$

c.

$$\frac{2\times 5}{(s+5)^2+5^2} = \frac{10}{(s+5)^2+25}$$

d.

$$\frac{8(s+3)}{(s+3)^2+4^2} = \frac{8(s+3)}{(s+3)^2+16}$$

e.

$$\cos t\big|_{\pi/4}\delta(t-\pi/4) = (\sqrt{2}/2)\delta(t-\pi/4) \text{ and } (\sqrt{2}/2)\delta(t-\pi/4) \Leftrightarrow (\sqrt{2}/2)e^{-(\pi/4)s}$$

5.

a.

$$5tu_0(t-3) = [5(t-3)+15]u_0(t-3) \Leftrightarrow e^{-3s}\left(\frac{5}{s^2}+\frac{15}{s}\right) = \frac{5}{s}e^{-3s}\left(\frac{1}{s}+3\right)$$

b.

$$(2t^2-5t+4)u_0(t-3) = [2(t-3)^2+12t-18-5t+4]u_0(t-3)$$

$$= [2(t-3)^2+7t-14]u_0(t-3)$$

$$= [2(t-3)^2+7(t-3)+21-14]u_0(t-3)$$

$$= [2(t-3)^2+7(t-3)+7]u_0(t-3) \Leftrightarrow e^{-3s}\left(\frac{2\times 2!}{s^3}+\frac{7}{s^2}+\frac{7}{s}\right)$$

c.

$$(t-3)e^{-2t}u_0(t-2) = [(t-2)-1]e^{-2(t-2)}\cdot e^{-4}u_0(t-2)$$

$$\Leftrightarrow e^{-4}\cdot e^{-2s}\left[\frac{1}{(s+2)^2}-\frac{1}{(s+2)}\right] = e^{-4}\cdot e^{-2s}\left[\frac{-(s+1)}{(s+2)^2}\right]$$

d.

$$(2t-4)e^{2(t-2)}u_0(t-3) = [2(t-3)+6-4]e^{-2(t-3)}\cdot e^{-2}u_0(t-3)$$

$$\Leftrightarrow e^{-2}\cdot e^{-3s}\left[\frac{2}{(s+3)^2}+\frac{2}{(s+3)}\right] = 2e^{-2}\cdot e^{-3s}\left[\frac{s+4}{(s+3)^2}\right]$$

e.

$$4te^{-3t}(\cos 2t)u_0(t) \Leftrightarrow 4(-1)^1\frac{d}{ds}\left[\frac{s+3}{(s+3)^2+2^2}\right] = -4\frac{d}{ds}\left[\frac{s+3}{s^2+6s+9+4}\right]$$

$$\Leftrightarrow -4\frac{d}{ds}\left[\frac{s+3}{s^2+6s+13}\right] = -4\left[\frac{s^2+6s+13-(s+3)(2s+6)}{(s^2+6s+13)^2}\right]$$

$$\Leftrightarrow -4\left[\frac{s^2+6s+13-2s^2-6s-6s-18}{(s^2+6s+13)^2}\right] = \frac{4(s^2+6s+5)}{(s^2+6s+13)^2}$$

6.

a.

$$\sin 3t \Leftrightarrow \frac{3}{s^2+3^2} \qquad \frac{d}{dt}f(t) \Leftrightarrow sF(s)-f(0^-) \qquad f(0^-) = \sin 3t\big|_{t=0} = 0$$

$$\frac{d}{dt}(\sin 3t) \Leftrightarrow s\frac{3}{s^2+3^2}-0 = \frac{3s}{s^2+9}$$

b.

$$3e^{-4t} \Leftrightarrow \frac{3}{s+4} \qquad \frac{d}{dt}f(t) \Leftrightarrow sF(s)-f(0^-) \qquad f(0^-) = 3e^{-4t}\big|_{t=0} = 3$$

$$\frac{d}{dt}(3e^{-4t}) \Leftrightarrow s\frac{3}{s+4}-3 = \frac{3s}{s+4}-\frac{3(s+4)}{s+4} = \frac{-12}{s+4}$$

c.

$$\cos 2t \Leftrightarrow \frac{s}{s^2+2^2} \qquad t^2\cos 2t \Leftrightarrow (-1)^2\frac{d^2}{ds^2}\left[\frac{s}{s^2+4}\right]$$

$$\frac{d}{ds}\left[\frac{s^2+4-s(2s)}{(s^2+4)^2}\right] = \frac{d}{ds}\left[\frac{-s^2+4}{(s^2+4)^2}\right] = \frac{(s^2+4)^2(-2s)-(-s^2+4)(s^2+4)2(2s)}{(s^2+4)^4}$$

$$= \frac{(s^2+4)(-2s)-(-s^2+4)(4s)}{(s^2+4)^3} = \frac{-2s^3-8s+4s^3-16s}{(s^2+4)^3} = \frac{2s(s^2-12)}{(s^2+4)^3}$$

Thus,

$$t^2\cos 2t \Leftrightarrow \frac{2s(s^2-12)}{(s^2+4)^3}$$

and

$$\frac{d}{dt}(t^2\cos 2t) \Leftrightarrow sF(s) - f(0^-)$$

$$\Leftrightarrow s\frac{2s(s^2-12)}{(s^2+4)^3} - 0 = \frac{2s^2(s^2-12)}{(s^2+4)^3}$$

d.

$$\sin 2t \Leftrightarrow \frac{2}{s^2+2^2} \qquad e^{-2t}\sin 2t \Leftrightarrow \frac{2}{(s+2)^2+4} \qquad \frac{d}{dt}f(t) \Leftrightarrow sF(s) - f(0^-)$$

$$\frac{d}{dt}(e^{-2t}\sin 2t) \Leftrightarrow s\frac{2}{(s+2)^2+4} - 0 = \frac{2s}{(s+2)^2+4}$$

e.

$$t^2 \Leftrightarrow \frac{2!}{s^3} \qquad t^2e^{-2t} \Leftrightarrow \frac{2!}{(s+2)^3} \qquad \frac{d}{dt}f(t) \Leftrightarrow sF(s) - f(0^-)$$

$$\frac{d}{dt}(t^2e^{-2t}) \Leftrightarrow s\frac{2!}{(s+2)^3} - 0 = \frac{2s}{(s+2)^3}$$

7.

a.

$\sin t \Leftrightarrow \frac{1}{s^2+1}$ but to find $\mathcal{L}\left\{\frac{\sin t}{t}\right\}$ we must first show that the limit $\lim_{t\to 0}\frac{\sin t}{t}$ exists. Since $\lim_{x\to 0}\frac{\sin x}{x} = 1$, this condition is satisfied and thus $\frac{\sin t}{t} \Leftrightarrow \int_s^\infty \frac{1}{s^2+1}ds$. From tables of integrals, $\int\frac{1}{x^2+a^2}dx = \frac{1}{a}\tan^{-1}(x/a) + C$. Then, $\int\frac{1}{s^2+1}ds = \tan^{-1}(1/s) + C$ and the constant of integration C is evaluated from the final value theorem. Thus,

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s) = \lim_{s\to 0} s[\tan^{-1}(1/s) + C] = 0 \text{ and } \frac{\sin t}{t} \Leftrightarrow \tan^{-1}(1/s)$$

b.

From (a) above, $\frac{\sin t}{t} \Leftrightarrow \tan^{-1}(1/s)$ and since $\int_{-\infty}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(s)}{s} + \frac{f(0^-)}{s}$, it follows that

$$\int_0^t \frac{\sin\tau}{\tau}d\tau \Leftrightarrow \frac{1}{s}\tan^{-1}(1/s)$$

c.

From (a) above $\frac{\sin t}{t} \Leftrightarrow \tan^{-1}(1/s)$ and since $f(at) \Leftrightarrow \frac{1}{a}F\left(\frac{s}{a}\right)$, it follows that

$$\frac{\sin at}{at} \Leftrightarrow \frac{1}{a}\tan^{-1}\left(\frac{1/s}{a}\right) \text{ or } \frac{\sin at}{t} \Leftrightarrow \tan^{-1}(a/s)$$

d.

$\cos t \Leftrightarrow \dfrac{s}{s^2+1}$, $\dfrac{\cos t}{t} \Leftrightarrow \displaystyle\int_s^\infty \frac{s}{s^2+1}ds$, and from tables of integrals,

$\displaystyle\int \frac{x}{x^2+a^2}dx = \frac{1}{2}\ln(x^2+a^2)+C$. Then, $\displaystyle\int \frac{s}{s^2+1}ds = \frac{1}{2}\ln(s^2+1)+C$ and the constant of integration C is evaluated from the final value theorem. Thus,

$\displaystyle\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s) = \lim_{s\to 0} s\left[\frac{1}{2}\ln(s^2+1)+C\right] = 0$ and using $\displaystyle\int_{-\infty}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(s)}{s} + \frac{f(0^-)}{s}$ we obtain

$$\int_t^\infty \frac{\cos\tau}{\tau}d\tau \Leftrightarrow \frac{1}{2s}\ln(s^2+1)$$

e.

$e^{-t} \Leftrightarrow \dfrac{1}{s+1}$, $\dfrac{e^{-t}}{t} \Leftrightarrow \displaystyle\int_s^\infty \frac{1}{s+1}ds$, and from tables of integrals $\displaystyle\int \frac{1}{ax+b}dx = \frac{1}{2}\ln(ax+b)$. Then,

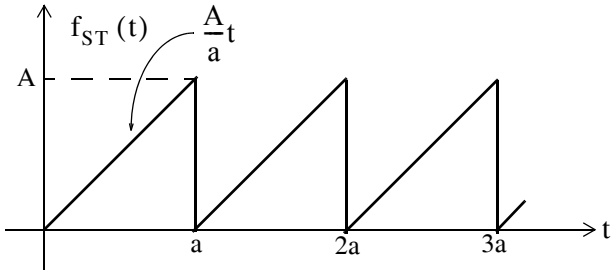
$\displaystyle\int \frac{1}{s+1}ds = \ln(s+1)+C$ and the constant of integration C is evaluated from the final value theorem. Thus,

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s) = \lim_{s\to 0} s[\ln(s+1)+C] = 0$$

and using $\displaystyle\int_{-\infty}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(s)}{s} + \frac{f(0^-)}{s}$, we obtain

$$\int_t^\infty \frac{e^{-\tau}}{\tau}d\tau \Leftrightarrow \frac{1}{s}\ln(s+1)$$

8.



This is a periodic waveform with period $T = a$, and its Laplace transform is

$$F(s) = \frac{1}{1-e^{-as}}\int_0^a \frac{A}{a}te^{-st}dt = \frac{A}{a(1-e^{-as})}\int_0^a te^{-st}dt \quad (1)$$

and from (2.41), Page 2-14, and limits of integration 0 to a, we obtain

$$\mathcal{L}\{t\}\Big|_0^a = \int_0^a te^{-st}dt = \left[-\frac{te^{-st}}{s} - \frac{e^{-st}}{s^2}\right]\Bigg|_0^a = \left[\frac{te^{-st}}{s} + \frac{e^{-st}}{s^2}\right]\Bigg|_a^0$$

$$= \left[\frac{1}{s^2} - \frac{ae^{-as}}{s} - \frac{e^{-as}}{s^2}\right] = \frac{1}{s^2}[1-(1+as)e^{-as}]$$

Adding and subtracting as in the last expression above, we obtain

$$\mathcal{L}\{t\}\Big|_0^a = \frac{1}{s^2}[(1+as)-(1+as)e^{-as}-as] = \frac{1}{s^2}[(1+as)(1-e^{-as})-as]$$

By substitution into (1) we obtain

$$F(s) = \frac{A}{a(1-e^{-as})}\cdot\frac{1}{s^2}[(1+as)(1-e^{-as})-as] = \frac{A}{as^2(1-e^{-as})}\cdot[(1+as)(1-e^{-as})-as]$$

$$= \frac{A(1+as)}{as^2} - \frac{Aa}{as(1-e^{-as})} = \frac{A}{as}\left[\frac{(1+as)}{s} - \frac{a}{(1-e^{-as})}\right]$$

9.

This is a periodic waveform with period $T = a = \pi$ and its Laplace transform is

$$F(s) = \frac{1}{1-e^{-sT}}\int_0^T f(t)e^{-st}dt = \frac{1}{(1-e^{-\pi s})}\int_0^\pi \sin te^{-st}dt$$

From tables of integrals,

$$\int \sin bx e^{ax}dx = \frac{e^{ax}(a\sin bx - b\cos bx)}{a^2+b^2}$$

Then,

$$F(s) = \frac{1}{1-e^{-\pi s}}\cdot\frac{e^{-st}(s\sin t - \cos t)}{s^2+1}\Bigg|_0^\pi = \frac{1}{1-e^{-\pi s}}\cdot\frac{1+e^{-\pi s}}{s^2+1}$$

$$= \frac{1}{s^2+1}\cdot\frac{1+e^{-\pi s}}{1-e^{-\pi s}} = \frac{1}{s^2+1}\coth\left(\frac{\pi s}{2}\right)$$

The full–rectified waveform can be produced with the MATLAB script

```
t=0:pi/16:4*pi; x=sin(t); plot(t,abs(x)); axis([0 4*pi 0 1])
```

Chapter 3

The Inverse Laplace Transformation

This chapter is a continuation to the Laplace transformation topic of the previous chapter and presents several methods of finding the Inverse Laplace Transformation. The partial fraction expansion method is explained thoroughly and it is illustrated with several examples.

3.1 The Inverse Laplace Transform Integral

The Inverse Laplace Transform Integral was stated in the previous chapter; it is repeated here for convenience.

$$\mathcal{L}^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F(s)e^{st}ds \tag{3.1}$$

This integral is difficult to evaluate because it requires contour integration using complex variables theory. Fortunately, for most engineering problems we can refer to Tables of Properties, and Common Laplace transform pairs to lookup the Inverse Laplace transform.

3.2 Partial Fraction Expansion

Quite often the Laplace transform expressions are not in recognizable form, but in most cases appear in a rational form of s, that is,

$$F(s) = \frac{N(s)}{D(s)} \tag{3.2}$$

where $N(s)$ and $D(s)$ are polynomials, and thus (3.2) can be expressed as

$$F(s) = \frac{N(s)}{D(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + b_{m-2}s^{m-2} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \ldots + a_1 s + a_0} \tag{3.3}$$

The coefficients a_k and b_k are real numbers for $k = 1, 2, \ldots, n$, and if the highest power m of $N(s)$ is less than the highest power n of $D(s)$, i.e., $m < n$, $F(s)$ is said to be expressed as a *proper rational function*. If $m \geq n$, $F(s)$ is an *improper rational function.*

In a proper rational function, the roots of $N(s)$ in (3.3) are found by setting $N(s) = 0$; these are called the *zeros* of $F(s)$. The roots of $D(s)$, found by setting $D(s) = 0$, are called the *poles* of $F(s)$. We assume that $F(s)$ in (3.3) is a proper rational function. Then, it is customary and very conve-

nient to make the coefficient of s^n unity; thus, we rewrite $F(s)$ as

$$F(s) = \frac{N(s)}{D(s)} = \frac{\frac{1}{a_n}(b_m s^m + b_{m-1}s^{m-1} + b_{m-2}s^{m-2} + \ldots + b_1 s + b_0)}{s^n + \frac{a_{n-1}}{a_n}s^{n-1} + \frac{a_{n-2}}{a_n}s^{n-2} + \ldots + \frac{a_1}{a_n}s + \frac{a_0}{a_n}} \tag{3.4}$$

The zeros and poles of (3.4) can be real and distinct, repeated, complex conjugates, or combinations of real and complex conjugates. However, we are mostly interested in the nature of the poles, so we will consider each case separately, as indicated in Subsections 3.2.1 through 3.2.3 below.

3.2.1 Distinct Poles

If all the poles $p_1, p_2, p_3, \ldots, p_n$ of $F(s)$ are *distinct* (different from each another), we can factor the denominator of $F(s)$ in the form

$$F(s) = \frac{N(s)}{(s-p_1)\cdot(s-p_2)\cdot(s-p_3)\cdot\ldots\cdot(s-p_n)} \tag{3.5}$$

where p_k is distinct from all other poles. Next, using the *partial fraction expansion method,** we can express (3.5) as

$$F(s) = \frac{r_1}{(s-p_1)} + \frac{r_2}{(s-p_2)} + \frac{r_3}{(s-p_3)} + \ldots + \frac{r_n}{(s-p_n)} \tag{3.6}$$

where $r_1, r_2, r_3, \ldots, r_n$ are the *residues*, and $p_1, p_2, p_3, \ldots, p_n$ are the *poles* of $F(s)$.

To evaluate the residue r_k, we multiply both sides of (3.6) by $(s-p_k)$; then, we let $s \rightarrow p_k$, that is,

$$r_k = \lim_{s \rightarrow p_k}(s-p_k)F(s) = (s-p_k)F(s)\Big|_{s=p_k} \tag{3.7}$$

Example 3.1

Use the partial fraction expansion method to simplify $F_1(s)$ of (3.8) below, and find the time domain function $f_1(t)$ corresponding to $F_1(s)$.

* *The partial fraction expansion method applies only to proper rational functions. It is used extensively in integration, and in finding the inverses of the Laplace transform, the Fourier transform, and the z-transform. This method allows us to decompose a rational polynomial into smaller rational polynomials with simpler denominators from which we can easily recognize their integrals and inverse transformations. This method is also being taught in intermediate algebra and introductory calculus courses.*

$$F_1(s) = \frac{3s+2}{s^2+3s+2} \tag{3.8}$$

Solution:

Using (3.6), we obtain

$$F_1(s) = \frac{3s+2}{s^2+3s+2} = \frac{3s+2}{(s+1)(s+2)} = \frac{r_1}{(s+1)} + \frac{r_2}{(s+2)} \tag{3.9}$$

The residues are

$$r_1 = \lim_{s \to -1}(s+1)F(s) = \left.\frac{3s+2}{(s+2)}\right|_{s=-1} = -1 \tag{3.10}$$

and

$$r_2 = \lim_{s \to -2}(s+2)F(s) = \left.\frac{3s+2}{(s+1)}\right|_{s=-2} = 4 \tag{3.11}$$

Therefore, we express (3.9) as

$$F_1(s) = \frac{3s+2}{s^2+3s+2} = \frac{-1}{(s+1)} + \frac{4}{(s+2)} \tag{3.12}$$

and from Table 2.2, Chapter 2, Page 2–22, we find that

$$e^{-at}u_0(t) \Leftrightarrow \frac{1}{s+a} \tag{3.13}$$

Therefore,

$$F_1(s) = \frac{-1}{(s+1)} + \frac{4}{(s+2)} \Leftrightarrow (-e^{-t} + 4e^{-2t})\,u_0(t) = f_1(t) \tag{3.14}$$

The residues and poles of a rational function of polynomials such as (3.8), can be found easily using the MATLAB **residue(a,b)** function. For this example, we use the script

```
Ns = [3, 2]; Ds = [1, 3, 2]; [r, p, k] = residue(Ns, Ds)
```

and MATLAB returns the values

```
r =
     4
    -1
p =
    -2
    -1
k =
     []
```

For the MATLAB script above, we defined Ns and Ds as two vectors that contain the numerator and denominator coefficients of F(s). When this script is executed, MATLAB displays the r and p vectors that represent the residues and poles respectively. The first value of the vector r is associated with the first value of the vector p, the second value of r is associated with the second value of p, and so on.

The vector k is referred to as the *direct term* and it is always empty (has no value) whenever F(s) is a proper rational function, that is, when the highest degree of the denominator is larger than that of the numerator. For this example, we observe that the highest power of the denominator is s^2, whereas the highest power of the numerator is s and therefore the direct term is empty.

We can also use the MATLAB **ilaplace(f)** function to obtain the time domain function directly from F(s). This is done with the script that follows.

```
syms s t; Fs=(3*s+2)/(s^2+3*s+2); ft=ilaplace(Fs); pretty(ft)
```

When this script is executed, MATLAB displays the expression

```
4 exp(-2 t)- exp(-t)
```

Example 3.2

Use the partial fraction expansion method to simplify $F_2(s)$ of (3.15) below, and find the time domain function $f_2(t)$ corresponding to $F_2(s)$.

$$F_2(s) = \frac{3s^2 + 2s + 5}{s^3 + 12s^2 + 44s + 48} \tag{3.15}$$

Solution:

First, we use the MATLAB **factor(s)** symbolic function to express the denominator polynomial of $F_2(s)$ in factored form. For this example,

```
syms s; factor(s^3 + 12*s^2 + 44*s + 48)
```

```
ans =
(s+2)*(s+4)*(s+6)
```

Then,

$$F_2(s) = \frac{3s^2 + 2s + 5}{s^3 + 12s^2 + 44s + 48} = \frac{3s^2 + 2s + 5}{(s+2)(s+4)(s+6)} = \frac{r_1}{(s+2)} + \frac{r_2}{(s+4)} + \frac{r_3}{(s+6)} \tag{3.16}$$

The residues are

$$r_1 = \left.\frac{3s^2 + 2s + 5}{(s+4)(s+6)}\right|_{s=-2} = \frac{9}{8} \tag{3.17}$$

$$r_2 = \left.\frac{3s^2 + 2s + 5}{(s+2)(s+6)}\right|_{s=-4} = -\frac{37}{4} \tag{3.18}$$

$$r_3 = \left.\frac{3s^2 + 2s + 5}{(s+2)(s+4)}\right|_{s=-6} = \frac{89}{8} \tag{3.19}$$

Then, by substitution into (3.16) we obtain

$$F_2(s) = \frac{3s^2 + 2s + 5}{s^3 + 12s^2 + 44s + 48} = \frac{9/8}{(s+2)} + \frac{-37/4}{(s+4)} + \frac{89/8}{(s+6)} \tag{3.20}$$

From Table 2.2, Chapter 2, Page 2–22,

$$e^{-at}u_0(t) \Leftrightarrow \frac{1}{s+a} \tag{3.21}$$

Therefore,

$$F_2(s) = \frac{9/8}{(s+2)} + \frac{-37/4}{(s+4)} + \frac{89/8}{(s+6)} \Leftrightarrow \left(\frac{9}{8}e^{-2t} - \frac{37}{4}e^{-4t} + \frac{89}{8}e^{-6t}\right)u_0(t) = f_2(t) \tag{3.22}$$

Check with MATLAB:

```
syms s t; Fs = (3*s^2 + 4*s + 5) / (s^3 + 12*s^2 + 44*s + 48); ft = ilaplace(Fs)

ft =
-37/4*exp(-4*t)+9/8*exp(-2*t)+89/8*exp(-6*t)
```

3.2.2 Complex Poles

Quite often, the poles of $F(s)$ are complex,* and since complex poles occur in complex conjugate pairs, the number of complex poles is even. Thus, if p_k is a complex root of $D(s)$, then, its complex conjugate pole, denoted as p_k^*, is also a root of $D(s)$. The partial fraction expansion method can also be used in this case, but it may be necessary to manipulate the terms of the expansion in order to express them in a recognizable form. The procedure is illustrated with the following example.

* *A review of complex numbers is presented in Appendix C*

Example 3.3

Use the partial fraction expansion method to simplify $F_3(s)$ of (3.23) below, and find the time domain function $f_3(t)$ corresponding to $F_3(s)$.

$$F_3(s) = \frac{s+3}{s^3+5s^2+12s+8} \tag{3.23}$$

Solution:

Let us first express the denominator in factored form to identify the poles of $F_3(s)$ using the MATLAB **factor(s)** symbolic function. Then,

```
syms s; factor(s^3 + 5*s^2 + 12*s + 8)
```

```
ans =
(s+1)*(s^2+4*s+8)
```

The **factor(s)** function did not factor the quadratic term. We will use the **roots(p)** function.

```
p=[1  4  8]; roots_p=roots(p)
```

```
roots_p =
  -2.0000 + 2.0000i
  -2.0000 - 2.0000i
```

Then,

$$F_3(s) = \frac{s+3}{s^3+5s^2+12s+8} = \frac{s+3}{(s+1)(s+2+j2)(s+2-j2)}$$

or

$$F_3(s) = \frac{s+3}{s^3+5s^2+12s+8} = \frac{r_1}{(s+1)} + \frac{r_2}{(s+2+j2)} + \frac{r_2^*}{(s+2-j2)} \tag{3.24}$$

The residues are

$$r_1 = \left.\frac{s+3}{s^2+4s+8}\right|_{s=-1} = \frac{2}{5} \tag{3.25}$$

$$\begin{aligned} r_2 &= \left.\frac{s+3}{(s+1)(s+2-j2)}\right|_{s=-2-j2} = \frac{1-j2}{(-1-j2)(-j4)} = \frac{1-j2}{-8+j4} \\ &= \frac{(1-j2)}{(-8+j4)}\frac{(-8-j4)}{(-8-j4)} = \frac{-16+j12}{80} = -\frac{1}{5}+\frac{j3}{20} \end{aligned} \tag{3.26}$$

$$r_2^* = \left(-\frac{1}{5}+\frac{j3}{20}\right)^* = -\frac{1}{5}-\frac{j3}{20} \tag{3.27}$$

By substitution into (3.24),

$$F_3(s) = \frac{2/5}{(s+1)} + \frac{-1/5 + j3/20}{(s+2+j2)} + \frac{-1/5 - j3/20}{(s+2-j2)} \tag{3.28}$$

The last two terms on the right side of (3.28), do not resemble any Laplace transform pair that we derived in Chapter 2. Therefore, we will express them in a different form. We combine them into a single term*, and now (3.28) is written as

$$F_3(s) = \frac{2/5}{(s+1)} - \frac{1}{5} \cdot \frac{(2s+1)}{(s^2+4s+8)} \tag{3.29}$$

For convenience, we denote the first term on the right side of (3.29) as $F_{31}(s)$, and the second as $F_{32}(s)$. Then,

$$F_{31}(s) = \frac{2/5}{(s+1)} \Leftrightarrow \frac{2}{5}e^{-t} = f_{31}(t) \tag{3.30}$$

Next, for $F_{32}(s)$

$$F_{32}(s) = -\frac{1}{5} \cdot \frac{(2s+1)}{(s^2+4s+8)} \tag{3.31}$$

From Table 2.2, Chapter 2, Page 2–22,

$$\begin{aligned} e^{-at}\sin\omega t u_0 t &\Leftrightarrow \frac{\omega}{(s+a)^2+\omega^2} \\ e^{-at}\cos\omega t u_0 t &\Leftrightarrow \frac{s+a}{(s+a)^2+\omega^2} \end{aligned} \tag{3.32}$$

Accordingly, we express $F_{32}(s)$ as

$$\begin{aligned} F_{32}(s) &= -\frac{2}{5}\left(\frac{s+\frac{1}{2}+\frac{3}{2}-\frac{3}{2}}{(s+2)^2+2^2)}\right) = -\frac{2}{5}\left(\frac{s+2}{(s+2)^2+2^2)} + \frac{-3/2}{(s+2)^2+2^2)}\right) \\ &= -\frac{2}{5}\left(\frac{s+2}{(s+2)^2+2^2)}\right) + \frac{6/10}{2}\left(\frac{2}{(s+2)^2+2^2)}\right) \\ &= -\frac{2}{5}\left(\frac{s+2}{(s+2)^2+2^2)}\right) + \frac{3}{10}\left(\frac{2}{(s+2)^2+2^2)}\right) \end{aligned} \tag{3.33}$$

Addition of (3.30) with (3.33) yields

---

* *Here, we used MATLAB function* simple((−1/5 +3j/20)/(s+2+2j)+(−1/5 −3j/20)/(s+2−2j)). *The* **simple** *function, after several simplification tools that were displayed on the screen, returned* `(-2*s-1)/(5*s^2+20*s+40)`.

$$F_3(s) = F_{31}(s) + F_{32}(s) = \frac{2/5}{(s+1)} - \frac{2}{5}\left(\frac{s+2}{(s+2)^2+2^2)}\right) + \frac{3}{10}\left(\frac{2}{(s+2)^2+2^2)}\right)$$

$$\Leftrightarrow \frac{2}{5}e^{-t} - \frac{2}{5}e^{-2t}\cos 2t + \frac{3}{10}e^{-2t}\sin 2t = f_3(t)$$

Check with MATLAB:

```
syms a s t w;              % Define several symbolic variables
Fs=(s + 3)/(s^3 + 5*s^2 + 12*s + 8); ft=ilaplace(Fs)

ft =
2/5*exp(-t)-2/5*exp(-2*t)*cos(2*t)
+3/10*exp(-2*t)*sin(2*t)
```

3.2.3 Multiple (Repeated) Poles

In this case, $F(s)$ has simple poles, but one of the poles, say p_1, has a multiplicity m. For this condition, we express it as

$$F(s) = \frac{N(s)}{(s-p_1)^m(s-p_2)\ldots(s-p_{n-1})(s-p_n)} \tag{3.34}$$

Denoting the m residues corresponding to multiple pole p_1 as $r_{11}, r_{12}, r_{13}, \ldots, r_{1m}$, the partial fraction expansion of (3.34) is expressed as

$$\begin{aligned} F(s) = {} & \frac{r_{11}}{(s-p_1)^m} + \frac{r_{12}}{(s-p_1)^{m-1}} + \frac{r_{13}}{(s-p_1)^{m-2}} + \ldots + \frac{r_{1m}}{(s-p_1)} \\ & + \frac{r_2}{(s-p_2)} + \frac{r_3}{(s-p_3)} + \ldots + \frac{r_n}{(s-p_n)} \end{aligned} \tag{3.35}$$

For the simple poles $p_1, p_2, \ldots, p_n$, we proceed as before, that is, we find the residues from

$$r_k = \lim_{s \to p_k}(s-p_k)F(s) = (s-p_k)F(s)\Big|_{s=p_k} \tag{3.36}$$

The residues $r_{11}, r_{12}, r_{13}, \ldots, r_{1m}$ corresponding to the repeated poles, are found by multiplication of both sides of (3.35) by $(s-p)^m$. Then,

$$\begin{aligned} (s-p_1)^m F(s) = {} & r_{11} + (s-p_1)r_{12} + (s-p_1)^2 r_{13} + \ldots + (s-p_1)^{m-1} r_{1m} \\ & + (s-p_1)^m\left(\frac{r_2}{(s-p_2)} + \frac{r_3}{(s-p_3)} + \ldots + \frac{r_n}{(s-p_n)}\right) \end{aligned} \tag{3.37}$$

Next, taking the limit as $s \to p_1$ on both sides of (3.37), we obtain

$$\lim_{s \to p_1} (s-p_1)^m F(s) = r_{11} + \lim_{s \to p_1} [(s-p_1)r_{12} + (s-p_1)^2 r_{13} + \ldots + (s-p_1)^{m-1} r_{1m}]$$

$$+ \lim_{s \to p_1} \left[(s-p_1)^m \left(\frac{r_2}{(s-p_2)} + \frac{r_3}{(s-p_3)} + \ldots + \frac{r_n}{(s-p_n)} \right) \right]$$

or

$$r_{11} = \lim_{s \to p_1} (s-p_1)^m F(s) \tag{3.38}$$

and thus (3.38) yields the residue of the first repeated pole.

The residue r_{12} for the second repeated pole p_1, is found by differentiating (3.37) with respect to s and again, we let $s \to p_1$, that is,

$$r_{12} = \lim_{s \to p_1} \frac{d}{ds} [(s-p_1)^m F(s)] \tag{3.39}$$

In general, the residue r_{1k} can be found from

$$(s-p_1)^m F(s) = r_{11} + r_{12}(s-p_1) + r_{13}(s-p_1)^2 + \ldots \tag{3.40}$$

whose $(m-1)$th derivative of both sides is

$$(k-1)! r_{1k} = \lim_{s \to p_1} \frac{1}{(k-1)!} \frac{d^{k-1}}{ds^{k-1}} [(s-p_1)^m F(s)] \tag{3.41}$$

or

$$r_{1k} = \lim_{s \to p_1} \frac{1}{(k-1)!} \frac{d^{k-1}}{ds^{k-1}} [(s-p_1)^m F(s)] \tag{3.42}$$

Example 3.4

Use the partial fraction expansion method to simplify $F_4(s)$ of (3.43) below, and find the time domain function $f_4(t)$ corresponding to $F_4(s)$.

$$F_4(s) = \frac{s+3}{(s+2)(s+1)^2} \tag{3.43}$$

Solution:

We observe that there is a pole of multiplicity 2 at $s = -1$, and thus in partial fraction expansion form, $F_4(s)$ is written as

$$F_4(s) = \frac{s+3}{(s+2)(s+1)^2} = \frac{r_1}{(s+2)} + \frac{r_{21}}{(s+1)^2} + \frac{r_{22}}{(s+1)} \tag{3.44}$$

The residues are

$$r_1 = \left.\frac{s+3}{(s+1)^2}\right|_{s=-2} = 1$$

$$r_{21} = \left.\frac{s+3}{s+2}\right|_{s=-1} = 2$$

$$r_{22} = \left.\frac{d}{ds}\left(\frac{s+3}{s+2}\right)\right|_{s=-1} = \left.\frac{(s+2)-(s+3)}{(s+2)^2}\right|_{s=-1} = -1$$

The value of the residue r_{22} can also be found without differentiation as follows:

Substitution of the already known values of r_1 and r_{21} into (3.44), and letting $s = 0$ *, we obtain

$$\left.\frac{s+3}{(s+1)^2(s+2)}\right|_{s=0} = \left.\frac{1}{(s+2)}\right|_{s=0} + \left.\frac{2}{(s+1)^2}\right|_{s=0} + \left.\frac{r_{22}}{(s+1)}\right|_{s=0}$$

or

$$\frac{3}{2} = \frac{1}{2} + 2 + r_{22}$$

from which $r_{22} = -1$ as before. Finally,

$$F_4(s) = \frac{s+3}{(s+2)(s+1)^2} = \frac{1}{(s+2)} + \frac{2}{(s+1)^2} + \frac{-1}{(s+1)} \Leftrightarrow e^{-2t} + 2te^{-t} - e^{-t} = f_4(t) \tag{3.45}$$

Check with MATLAB:

```
syms s t; Fs=(s+3)/((s+2)*(s+1)^2); ft=ilaplace(Fs)
```

```
ft = exp(-2*t)+2*t*exp(-t)-exp(-t)
```

We can use the following script to check the partial fraction expansion.

```
syms s
Ns = [1  3];              % Coefficients of the numerator N(s) of F(s)
expand((s + 1)^2);        % Expands (s + 1)^2 to s^2 + 2*s + 1;
d1 = [1  2  1];           % Coefficients of (s + 1)^2 = s^2 + 2*s + 1 term in D(s)
d2 = [0  1  2];           % Coefficients of (s + 2) term in D(s)
Ds=conv(d1,d2);           % Multiplies polynomials d1 and d2 to express the
                          % denominator D(s) of F(s) as a polynomial
[r,p,k]=residue(Ns,Ds)
```

* *This is permissible since (3.44) is an identity.*

```
r =
    1.0000
   -1.0000
    2.0000
p =
   -2.0000
   -1.0000
   -1.0000
k =
     []
```

Example 3.5

Use the partial fraction expansion method to simplify $F_5(s)$ of (3.46) below, and find the time domain function $f_5(t)$ corresponding to the given $F_5(s)$.

$$F_5(s) = \frac{s^2 + 3s + 1}{(s+1)^3(s+2)^2} \tag{3.46}$$

Solution:

We observe that there is a pole of multiplicity *3* at $s = -1$, and a pole of multiplicity 2 at $s = -2$. Then, in partial fraction expansion form, $F_5(s)$ is written as

$$F_5(s) = \frac{r_{11}}{(s+1)^3} + \frac{r_{12}}{(s+1)^2} + \frac{r_{13}}{(s+1)} + \frac{r_{21}}{(s+2)^2} + \frac{r_{22}}{(s+2)} \tag{3.47}$$

The residues are

$$r_{11} = \left.\frac{s^2 + 3s + 1}{(s+2)^2}\right|_{s=-1} = -1$$

$$r_{12} = \left.\frac{d}{ds}\left(\frac{s^2 + 3s + 1}{(s+2)^2}\right)\right|_{s=-1}$$

$$= \left.\frac{(s+2)^2(2s+3) - 2(s+2)(s^2+3s+1)}{(s+2)^4}\right|_{s=-1} = \left.\frac{s+4}{(s+2)^3}\right|_{s=-1} = 3$$

$$r_{13} = \frac{1}{2!}\frac{d^2}{ds^2}\left(\frac{s^2+3s+1}{(s+2)^2}\right)\bigg|_{s=-1} = \frac{1}{2}\frac{d}{ds}\left[\frac{d}{ds}\left(\frac{s^2+3s+1}{(s+2)^2}\right)\right]\bigg|_{s=-1}$$

$$= \frac{1}{2}\frac{d}{ds}\left(\frac{s+4}{(s+2)^3}\right)\bigg|_{s=-1} = \frac{1}{2}\left[\frac{(s+2)^3-3(s+2)^2(s+4)}{(s+2)^6}\right]\bigg|_{s=-1}$$

$$= \frac{1}{2}\left(\frac{s+2-3s-12}{(s+2)^4}\right)\bigg|_{s=-1} = \frac{-s-5}{(s+2)^4}\bigg|_{s=-1} = -4$$

Next, for the pole at $s = -2$,

$$r_{21} = \frac{s^2+3s+1}{(s+1)^3}\bigg|_{s=-2} = 1$$

and

$$r_{22} = \frac{d}{ds}\left(\frac{s^2+3s+1}{(s+1)^3}\right)\bigg|_{s=-2} = \frac{(s+1)^3(2s+3)-3(s+1)^2(s^2+3s+1)}{(s+1)^6}\bigg|_{s=-2}$$

$$= \frac{(s+1)(2s+3)-3(s^2+3s+1)}{(s+1)^4}\bigg|_{s=-2} = \frac{-s^2-4s}{(s+1)^4}\bigg|_{s=-2} = 4$$

By substitution of the residues into (3.47), we obtain

$$F_5(s) = \frac{-1}{(s+1)^3} + \frac{3}{(s+1)^2} + \frac{-4}{(s+1)} + \frac{1}{(s+2)^2} + \frac{4}{(s+2)} \tag{3.48}$$

We will check the values of these residues with the MATLAB script below.

```
syms s;     % The function collect(s) below multiplies (s+1)^3 by (s+2)^2
            % and we use it to express the denominator D(s) as a polynomial so that we can
            % use the coefficients of the resulting polynomial with the residue function
Ds=collect(((s+1)^3)*((s+2)^2))
```

```
Ds =
s^5+7*s^4+19*s^3+25*s^2+16*s+4
```

```
Ns=[1 3 1];  Ds=[1 7 19 25 16 4];  [r,p,k]=residue(Ns,Ds)
r =
    4.0000
    1.0000
   -4.0000
    3.0000
   -1.0000
```

```
p =
   -2.0000
   -2.0000
   -1.0000
   -1.0000
   -1.0000
k =
     []
```

From Table 2.2, Chapter 2, Page 2–22,

$$e^{-at} \Leftrightarrow \frac{1}{s+a} \qquad te^{-at} \Leftrightarrow \frac{1}{(s+a)^2} \qquad t^{n-1}e^{-at} \Leftrightarrow \frac{(n-1)!}{(s+a)^n}$$

and with these, we derive $f_5(t)$ from (3.48) as

$$f_5(t) = -\frac{1}{2}t^2e^{-t} + 3te^{-t} - 4e^{-t} + te^{-2t} + 4e^{-2t} \tag{3.49}$$

We can verify (3.49) with MATLAB as follows:

```
syms s t; Fs=-1/((s+1)^3) + 3/((s+1)^2) - 4/(s+1) + 1/((s+2)^2) + 4/(s+2); ft=ilaplace(Fs)
```

```
ft = -1/2*t^2*exp(-t)+3*t*exp(-t)-4*exp(-t)
          +t*exp(-2*t)+4*exp(-2*t)
```

3.3 Case where F(s) is Improper Rational Function

Our discussion thus far, was based on the condition that $F(s)$ is a proper rational function. However, if $F(s)$ is an improper rational function, that is, if $m \geq n$, we must first divide the numerator $N(s)$ by the denominator $D(s)$ to obtain an expression of the form

$$F(s) = k_0 + k_1 s + k_2 s^2 + \ldots + k_{m-n}s^{m-n} + \frac{N(s)}{D(s)} \tag{3.50}$$

where $N(s)/D(s)$ is a proper rational function.

Example 3.6

Derive the Inverse Laplace transform $f_6(t)$ of

$$F_6(s) = \frac{s^2 + 2s + 2}{s + 1} \tag{3.51}$$

Solution:

For this example, $F_6(s)$ is an improper rational function. Therefore, we must express it in the form of (3.50) before we use the partial fraction expansion method.

By long division, we obtain

$$F_6(s) = \frac{s^2 + 2s + 2}{s + 1} = \frac{1}{s + 1} + 1 + s$$

Now, we recognize that

$$\frac{1}{s + 1} \Leftrightarrow e^{-t}$$

and

$$1 \Leftrightarrow \delta(t)$$

but

$$s \Leftrightarrow ?$$

To answer that question, we recall that

$$u_0'(t) = \delta(t)$$

and

$$u_0''(t) = \delta'(t)$$

where $\delta'(t)$ is the doublet of the delta function. Also, by the time differentiation property

$$u_0''(t) = \delta'(t) \Leftrightarrow s^2F(s) - sf(0) - f'(0) = s^2F(s) = s^2 \cdot \frac{1}{s} = s$$

Therefore, we have the new transform pair

$$s \Leftrightarrow \delta'(t) \tag{3.52}$$

and thus,

$$F_6(s) = \frac{s^2 + 2s + 2}{s + 1} = \frac{1}{s + 1} + 1 + s \Leftrightarrow e^{-t} + \delta(t) + \delta'(t) = f_6(t) \tag{3.53}$$

In general,

$$\frac{d^n}{dt^n}\delta(t) \Leftrightarrow s^n \tag{3.54}$$

We verify (3.53) with MATLAB as follows:

```
Ns = [1  2  2]; Ds = [1  1]; [r, p, k] = residue(Ns, Ds)

r =
     1
p =
    -1
```

```
k =
     1     1
```

The direct terms `k= [1 1]` above are the coefficients of $\delta(t)$ and $\delta'(t)$ respectively.

3.4 Alternate Method of Partial Fraction Expansion

Partial fraction expansion can also be performed with the *method of clearing the fractions*, that is, making the denominators of both sides the same, then equating the numerators. As before, we assume that $F(s)$ is a proper rational function. If not, we first perform a long division, and then work with the quotient and the remainder as we did in Example 3.6. We also assume that the denominator $D(s)$ can be expressed as a product of real linear and quadratic factors. If these assumptions prevail, we let $(s-a)$ be a linear factor of $D(s)$, and we assume that $(s-a)^m$ is the highest power of $(s-a)$ that divides $D(s)$. Then, we can express $F(s)$ as

$$F(s) = \frac{N(s)}{D(s)} = \frac{r_1}{s-a} + \frac{r_2}{(s-a)^2} + \ldots \frac{r_m}{(s-a)^m} \tag{3.55}$$

Let $s^2+\alpha s+\beta$ be a quadratic factor of $D(s)$, and suppose that $(s^2+\alpha s+\beta)^n$ is the highest power of this factor that divides $D(s)$. Now, we perform the following steps:

1. To this factor, we assign the sum of *n* partial fractions, that is,

$$\frac{r_1 s + k_1}{s^2+\alpha s+\beta} + \frac{r_2 s + k_2}{(s^2+\alpha s+\beta)^2} + \ldots + \frac{r_n s + k_n}{(s^2+\alpha s+\beta)^n}$$

2. We repeat step 1 for each of the distinct linear and quadratic factors of $D(s)$
3. We set the given $F(s)$ equal to the sum of these partial fractions
4. We clear the resulting expression of fractions and arrange the terms in decreasing powers of s
5. We equate the coefficients of corresponding powers of s
6. We solve the resulting equations for the residues

Example 3.7

Express $F_7(s)$ of (3.56) below as a sum of partial fractions using the method of clearing the fractions.

$$F_7(s) = \frac{-2s+4}{(s^2+1)(s-1)^2} \tag{3.56}$$

Solution:

Using Steps 1 through 3 above, we obtain

$$F_7(s) = \frac{-2s+4}{(s^2+1)(s-1)^2} = \frac{r_1 s + A}{(s^2+1)} + \frac{r_{21}}{(s-1)^2} + \frac{r_{22}}{(s-1)} \tag{3.57}$$

With Step 4,

$$-2s+4 = (r_1 s + A)(s-1)^2 + r_{21}(s^2+1) + r_{22}(s-1)(s^2+1) \tag{3.58}$$

and with Step 5,

$$\begin{aligned} -2s+4 = {} & (r_1 + r_{22})s^3 + (-2r_1 + A - r_{22} + r_{21})s^2 \\ & + (r_1 - 2A + r_{22})s + (A - r_{22} + r_{21}) \end{aligned} \tag{3.59}$$

Relation (3.59) will be an identity is s if each power of s is the same on both sides of this relation. Therefore, we equate like powers of s and we obtain

$$\begin{aligned} 0 &= r_1 + r_{22} \\ 0 &= -2r_1 + A - r_{22} + r_{21} \\ -2 &= r_1 - 2A + r_{22} \\ 4 &= A - r_{22} + r_{21} \end{aligned} \tag{3.60}$$

Subtracting the second equation of (3.60) from the fourth, we obtain

$$4 = 2r_1$$

or

$$r_1 = 2 \tag{3.61}$$

By substitution of (3.61) into the first equation of (3.60), we obtain

$$0 = 2 + r_{22}$$

or

$$r_{22} = -2 \tag{3.62}$$

Next, substitution of (3.61) and (3.62) into the third equation of (3.60) yields

$$-2 = 2 - 2A - 2$$

or

$$A = 1 \tag{3.63}$$

Finally by substitution of (3.61), (3.62), and (3.63) into the fourth equation of (3.60), we obtain

$$4 = 1 + 2 + r_{21}$$

or

$$r_{21} = 1 \tag{3.64}$$

Substitution of these values into (3.57) yields

$$F_7(s) = \frac{-2s+4}{(s^2+1)(s-1)^2} = \frac{2s+1}{(s^2+1)} + \frac{1}{(s-1)^2} - \frac{2}{(s-1)} \tag{3.65}$$

Example 3.8

Use partial fraction expansion to simplify $F_8(s)$ of (3.66) below, and find the time domain function $f_8(t)$ corresponding to $F_8(s)$.

$$F_8(s) = \frac{s+3}{s^3+5s^2+12s+8} \tag{3.66}$$

Solution:

This is the same transform as in Example 3.3, Page 3–6, where we found that the denominator $D(s)$ can be expressed in factored form of a linear term and a quadratic. Thus, we write $F_8(s)$ as

$$F_8(s) = \frac{s+3}{(s+1)(s^2+4s+8)} \tag{3.67}$$

and using the method of clearing the fractions, we express (3.67) as

$$F_8(s) = \frac{s+3}{(s+1)(s^2+4s+8)} = \frac{r_1}{s+1} + \frac{r_2 s + r_3}{s^2+4s+8} \tag{3.68}$$

As in Example 3.3,

$$r_1 = \left.\frac{s+3}{s^2+4s+8}\right|_{s=-1} = \frac{2}{5} \tag{3.69}$$

Next, to compute r_2 and r_3, we follow the procedure of this section and we obtain

$$(s+3) = r_1(s^2+4s+8) + (r_2 s + r_3)(s+1) \tag{3.70}$$

Since r_1 is already known, we only need two equations in r_2 and r_3. Equating the coefficient of s^2 on the left side, which is zero, with the coefficients of s^2 on the right side of (3.70), we obtain

$$0 = r_1 + r_2 \tag{3.71}$$

and since $r_1 = 2/5$, it follows that $r_2 = -2/5$.

To obtain the third residue r_3, we equate the constant terms of (3.70). Then, $3 = 8r_1 + r_3$ or $3 = 8 \times 2/5 + r_3$, or $r_3 = -1/5$. Then, by substitution into (3.68), we obtain

$$F_8(s) = \frac{2/5}{(s+1)} - \frac{1}{5} \cdot \frac{(2s+1)}{(s^2+4s+8)} \tag{3.72}$$

as before.

The remaining steps are the same as in Example 3.3, and thus $f_8(t)$ is the same as $f_3(t)$, that is,

$$f_8(t) = f_3(t) = \left(\frac{2}{5}e^{-t} - \frac{2}{5}e^{-2t}\cos 2t + \frac{3}{10}e^{-2t}\sin 2t\right)u_0(t)$$

3.5 Summary

- The Inverse Laplace Transform Integral defined as

$$\mathcal{L}^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F(s)e^{st}ds$$

  is difficult to evaluate because it requires contour integration using complex variables theory.

- For most engineering problems we can refer to Tables of Properties, and Common Laplace transform pairs to lookup the Inverse Laplace transform.

- The partial fraction expansion method offers a convenient means of expressing Laplace transforms in a recognizable form from which we can obtain the equivalent time–domain functions.

- If the highest power m of the numerator $N(s)$ is less than the highest power n of the denominator $D(s)$, i.e., $m < n$, $F(s)$ is said to be expressed as a proper rational function. If $m \geq n$, $F(s)$ is an improper rational function.

- The Laplace transform $F(s)$ must be expressed as a proper rational function before applying the partial fraction expansion. If $F(s)$ is an improper rational function, that is, if $m \geq n$, we must first divide the numerator $N(s)$ by the denominator $D(s)$ to obtain an expression of the form

$$F(s) = k_0 + k_1 s + k_2 s^2 + \ldots + k_{m-n} s^{m-n} + \frac{N(s)}{D(s)}$$

- In a proper rational function, the roots of numerator $N(s)$ are called the zeros of $F(s)$ and the roots of the denominator $D(s)$ are called the poles of $F(s)$.

- The partial fraction expansion method can be applied whether the poles of $F(s)$ are distinct, complex conjugates, repeated, or a combination of these.

- When $F(s)$ is expressed as

$$F(s) = \frac{r_1}{(s-p_1)} + \frac{r_2}{(s-p_2)} + \frac{r_3}{(s-p_3)} + \ldots + \frac{r_n}{(s-p_n)}$$

  $r_1, r_2, r_3, \ldots, r_n$ are called the residues and $p_1, p_2, p_3, \ldots, p_n$ are the poles of $F(s)$.

- The residues and poles of a rational function of polynomials can be found easily using the MATLAB **residue(a,b)** function. The direct term is always empty (has no value) whenever $F(s)$ is a proper rational function.

- We can use the MATLAB **factor(s)** symbolic function to convert the denominator polynomial form of $F_2(s)$ into a factored form.

- We can use the MATLAB **collect(s)** and **expand(s)** symbolic functions to convert the denominator factored form of $F_2(s)$ into a polynomial form.

- In this chapter we introduced the new transform pair

$$s \Leftrightarrow \delta'(t)$$

and in general,

$$\frac{d^n}{dt^n}\delta(t) \Leftrightarrow s^n$$

- The method of clearing the fractions is an alternate method of partial fraction expansion.

3.6 Exercises

1. Find the Inverse Laplace transform of the following:

a. $\frac{4}{s+3}$ b. $\frac{4}{(s+3)^2}$ c. $\frac{4}{(s+3)^4}$ d. $\frac{3s+4}{(s+3)^5}$ e. $\frac{s^2+6s+3}{(s+3)^5}$

2. Find the Inverse Laplace transform of the following:

a. $\frac{3s+4}{s^2+4s+85}$ b. $\frac{4s+5}{s^2+5s+18.5}$ c. $\frac{s^2+3s+2}{s^3+5s^2+10.5s+9}$

d. $\frac{s^2-16}{s^3+8s^2+24s+32}$ e. $\frac{s+1}{s^3+6s^2+11s+6}$

3. Find the Inverse Laplace transform of the following:

a. $\frac{3s+2}{s^2+25}$ b. $\frac{5s^2+3}{(s^2+4)^2}$ Hint: $\left\{\begin{array}{l}\frac{1}{2\alpha}(\sin\alpha t+\alpha t\cos\alpha t)\Leftrightarrow\frac{s^2}{(s^2+\alpha^2)^2}\\ \frac{1}{2\alpha^3}(\sin\alpha t-\alpha t\cos\alpha t)\Leftrightarrow\frac{1}{(s^2+\alpha^2)^2}\end{array}\right\}$

c. $\frac{2s+3}{s^2+4.25s+1}$ d. $\frac{s^3+8s^2+24s+32}{s^2+6s+8}$ e. $e^{-2s}\frac{3}{(2s+3)^3}$

4. Use the Initial Value Theorem to find $f(0)$ given that the Laplace transform of $f(t)$ is

$$\frac{2s+3}{s^2+4.25s+1}$$

Compare your answer with that of Exercise 3(c).

5. It is known that the Laplace transform $F(s)$ has two distinct poles, one at $s=0$, the other at $s=-1$. It also has a single zero at $s=1$, and we know that $\lim_{t\to\infty} f(t)=10$. Find $F(s)$ and $f(t)$.

3.7 Solutions to End–of–Chapter Exercises

1.

a. $\dfrac{4}{s+3} \Leftrightarrow 4e^{-3t}$ b. $\dfrac{4}{(s+3)^2} \Leftrightarrow 4te^{-3t}$ c. $\dfrac{4}{(s+3)^4} \Leftrightarrow \dfrac{4}{3!}t^3e^{-3t} = \dfrac{2}{3}t^3e^{-3t}$

d.
$$\frac{3s+4}{(s+3)^5} = \frac{3(s+4/3+5/3-5/3)}{(s+3)^5} = 3\cdot\frac{(s+3)-5/3}{(s+3)^5} = 3\cdot\frac{1}{(s+3)^4} - 5\cdot\frac{1}{(s+3)^5}$$
$$\Leftrightarrow \frac{3}{3!}t^3e^{-3t} - \frac{5}{4!}t^4e^{-3t} = \frac{1}{2}\left(t^3e^{-3t} - \frac{5}{12}t^4e^{-3t}\right)$$

e.
$$\frac{s^2+6s+3}{(s+3)^5} = \frac{s^2+6s+9-6}{(s+3)^5} = \frac{(s+3)^2}{(s+3)^5} - \frac{6}{(s+3)^5} = \frac{1}{(s+3)^3} - 6\cdot\frac{1}{(s+3)^5}$$
$$\Leftrightarrow \frac{1}{2!}t^2e^{-3t} - \frac{6}{4!}t^4e^{-3t} = \frac{1}{2}\left(t^2e^{-3t} - \frac{1}{2}t^4e^{-3t}\right)$$

2.

a.
$$\frac{3s+4}{s^2+4s+85} = \frac{3(s+4/3+2/3-2/3)}{(s+2)^2+81} = 3\cdot\frac{(s+2)-2/3}{(s+2)^2+9^2} = 3\cdot\frac{(s+2)}{(s+2)^2+9^2} - \frac{1}{9}\cdot\frac{2\times 9}{(s+2)^2+9^2}$$
$$= 3\cdot\frac{(s+2)}{(s+2)^2+9^2} - \frac{2}{9}\cdot\frac{9}{(s+2)^2+9^2} \Leftrightarrow 3e^{-2t}\cos 9t - \frac{2}{9}e^{-2t}\sin 9t$$

b.
$$\frac{4s+5}{s^2+5s+18.5} = \frac{4s+5}{s^2+5s+6.25+12.25} = \frac{4s+5}{(s+2.5)^2+3.5^2} = 4\cdot\frac{s+5/4}{(s+2.5)^2+3.5^2}$$
$$= 4\cdot\frac{s+10/4-10/4+5/4}{(s+2.5)^2+3.5^2} = 4\cdot\frac{s+2.5}{(s+2.5)^2+3.5^2} - \frac{1}{3.5}\cdot\frac{5\times 3.5}{(s+2.5)^2+3.5^2}$$
$$= 4\cdot\frac{(s+2.5)}{(s+2.5)^2+3.5^2} - \frac{10}{7}\cdot\frac{3.5}{(s+2.5)^2+3.5^2} \Leftrightarrow 4e^{-2.5t}\cos 3.5t - \frac{10}{7}e^{-2.5t}\sin 3.5t$$

c. Using the MATLAB **factor(s)** function we obtain:

```
syms s; factor(s^2+3*s+2), factor(s^3+5*s^2+10.5*s+9)

ans = (s+2)*(s+1)
ans = 1/2*(s+2)*(2*s^2+6*s+9)
```

Then,

$$\frac{s^2+3s+2}{s^3+5s^2+10.5s+9}=\frac{(s+1)(s+2)}{(s+2)(s^2+3s+4.5)}=\frac{(s+1)}{(s^2+3s+4.5)}=\frac{s+1}{s^2+3s+2.25-2.25+4.5}$$

$$=\frac{s+1.5-1.5+1}{(s+1.5)^2+(1.5)^2}=\frac{s+1.5}{(s+1.5)^2+(1.5)^2}-\frac{1}{1.5}\cdot\frac{0.5\times 1.5}{(s+1.5)^2+(1.5)^2}$$

$$=\frac{s+1.5}{(s+1.5)^2+(1.5)^2}-\frac{1}{3}\cdot\frac{1.5}{(s+2.5)^2+3.5^2}\Leftrightarrow e^{-1.5t}\cos 1.5t-\frac{1}{3}e^{-1.5t}\sin 1.5t$$

d.

$$\frac{s^2-16}{s^3+8s^2+24s+32}=\frac{(s+4)(s-4)}{(s+4)(s^2+4s+8)}=\frac{(s-4)}{(s+2)^2+2^2}=\frac{s+2-2-4}{(s+2)^2+2^2}$$

$$=\frac{s+2}{(s+2)^2+2^2}-\frac{1}{2}\cdot\frac{6\times 2}{(s+2)^2+2^2}$$

$$=\frac{s+2}{(s+2)^2+2^2}-3\cdot\frac{2}{(s+2)^2+2^2}\Leftrightarrow e^{-2t}\cos 2t-3e^{-2t}\sin 2t$$

e.

$$\frac{s+1}{s^3+6s^2+11s+6}=\frac{(s+1)}{(s+1)(s+2)(s+3)}=\frac{1}{(s+2)(s+3)}$$

$$=\frac{1}{(s+2)(s+3)}=\frac{r_1}{s+2}+\frac{r_2}{s+3}\qquad r_1=\frac{1}{s+3}\bigg|_{s=-2}=1\qquad r_2=\frac{1}{s+2}\bigg|_{s=-3}=-1$$

$$=\frac{1}{(s+2)(s+3)}=\left[\frac{1}{s+2}-\frac{1}{s+3}\right]\Leftrightarrow e^{-2t}-e^{-3t}$$

3.

a. $$\frac{3s+2}{s^2+25}=\frac{3s}{s^2+5^2}+\frac{1}{5}\cdot\frac{2\times 5}{s^2+5^2}=3\cdot\frac{s}{s^2+5^2}+\frac{2}{5}\cdot\frac{5}{s^2+5^2}\Leftrightarrow 3\cos 5t+\frac{2}{5}\sin 5t$$

b. $$\frac{5s^2+3}{(s^2+4)^2}=\frac{5s^2}{(s^2+2^2)^2}+\frac{3}{(s^2+2^2)^2}\Leftrightarrow 5\cdot\frac{1}{2\times 2}(\sin 2t+2t\cos 2t)+3\cdot\frac{1}{2\times 8}(\sin 2t-2t\cos 2t)$$

$$\Leftrightarrow\left(\frac{5}{4}+\frac{3}{16}\right)\sin 2t+\left(\frac{5}{4}-\frac{3}{16}\right)2t\cos 2t=\frac{23}{16}\sin 2t+\frac{17}{8}t\cos 2t$$

c. $$\frac{2s+3}{s^2+4.25s+1}=\frac{2s+3}{(s+4)(s+1/4)}=\frac{r_1}{s+4}+\frac{r_2}{s+1/4}$$

$$r_1=\frac{2s+3}{s+1/4}\bigg|_{s=-4}=\frac{-5}{-15/4}=\frac{4}{3}\qquad r_2=\frac{2s+3}{s+4}\bigg|_{s=-1/4}=\frac{5/2}{15/4}=\frac{2}{3}$$

$$\frac{4/3}{s+4}+\frac{2/3}{s+1/4}\Leftrightarrow\frac{2}{3}(2e^{-4t}+e^{-t/4})$$

d. $\dfrac{s^3+8s^2+24s+32}{s^2+6s+8}=\dfrac{(s+4)(s^2+4s+8)}{(s+2)(s+4)}=\dfrac{(s^2+4s+8)}{(s+2)}$ and by long division

$$\frac{s^2+4s+8}{s+2}=s+2+\frac{4}{s+2}\Leftrightarrow\delta'(t)+2\delta(t)+4e^{-2t}$$

e.

$$e^{-2s}\frac{3}{(2s+3)^3}\qquad e^{-2s}F(s)\Leftrightarrow f(t-2)u_0(t-2)$$

$$F(s)=\frac{3}{(2s+3)^3}=\frac{3/2^3}{(2s+3)^3/2^3}=\frac{3/8}{[(2s+3)/2]^3}=\frac{3/8}{(s+3/2)^3}\Leftrightarrow\frac{3}{8}\left(\frac{1}{2!}t^2e^{-(3/2)t}\right)=\frac{3}{16}t^2e^{-(3/2)t}$$

$$e^{-2s}F(s)=e^{-2s}\frac{3}{(2s+3)^3}\Leftrightarrow\frac{3}{16}(t-2)^2e^{-(3/2)(t-2)}u_0(t-2)$$

4. The initial value theorem states that $\lim\limits_{t\to 0} f(t)=\lim\limits_{s\to\infty} sF(s)$. Then,

$$f(0)=\lim_{s\to\infty} s\frac{2s+3}{s^2+4.25s+1}=\lim_{s\to\infty}\frac{2s^2+3s}{s^2+4.25s+1}$$

$$=\lim_{s\to\infty}\frac{2s^2/s^2+3s/s^2}{s^2/s^2+4.25s/s^2+1/s^2}=\lim_{s\to\infty}\frac{2+3/s}{1+4.25/s+1/s^2}=2$$

The value $f(0)=2$ is the same as in the time domain expression that we found in Exercise 3(c).

5. We are given that $F(s)=\dfrac{A(s-1)}{s(s+1)}$ and $\lim\limits_{t\to\infty} f(t)=\lim\limits_{s\to 0} sF(s)=10$. Then,

$$\lim_{s\to 0} s\frac{A(s-1)}{s(s+1)}=A\lim_{s\to 0}\frac{(s-1)}{(s+1)}=-A=10$$

Therefore,

$$F(s)=\frac{-10(s-1)}{s(s+1)}=\frac{r_1}{s}+\frac{r_2}{s+1}=\frac{10}{s}-\frac{20}{s+1}\Leftrightarrow(10-20e^{-t})u_0(t)$$

that is,

$$f(t)=(10-20e^{-t})u_0(t)$$

and we observe that

$$\lim_{t\to\infty} f(t)=10$$

Chapter 4

Circuit Analysis with Laplace Transforms

This chapter presents applications of the Laplace transform. Several examples are presented to illustrate how the Laplace transformation is applied to circuit analysis. Complex impedance, complex admittance, and transfer functions are also defined.

4.1 Circuit Transformation from Time to Complex Frequency

In this section we will show the voltage–current relationships for the three elementary circuit networks, i.e., resistive, inductive, and capacitive in the time and complex frequency domains. They are shown in Subsections 4.1.1 through 4.1.3 below.

4.1.1 Resistive Network Transformation

The time and complex frequency domains for purely resistive networks are shown in Figure 4.1.

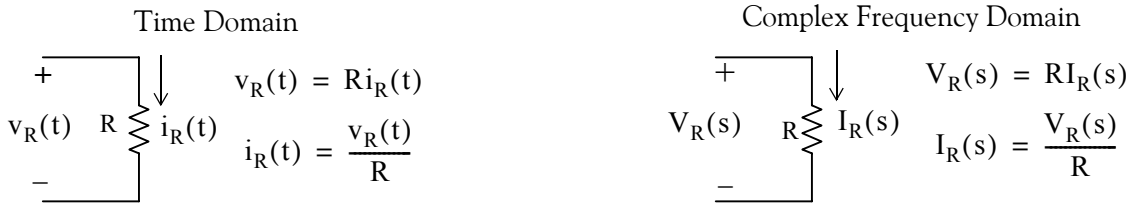


Figure 4.1. Resistive network in time domain and complex frequency domain

4.1.2 Inductive Network Transformation

The time and complex frequency domains for purely inductive networks are shown in Figure 4.2.

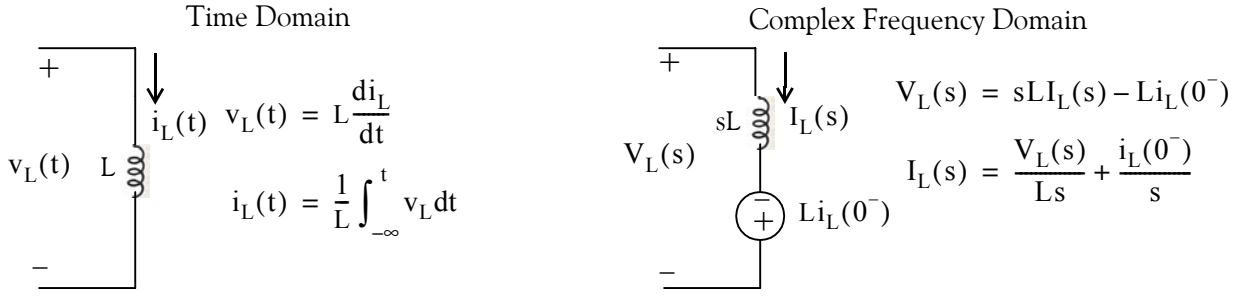


Figure 4.2. Inductive network in time domain and complex frequency domain

4.1.3 Capacitive Network Transformation

The time and complex frequency domains for purely capacitive networks are shown in Figure 4.3.

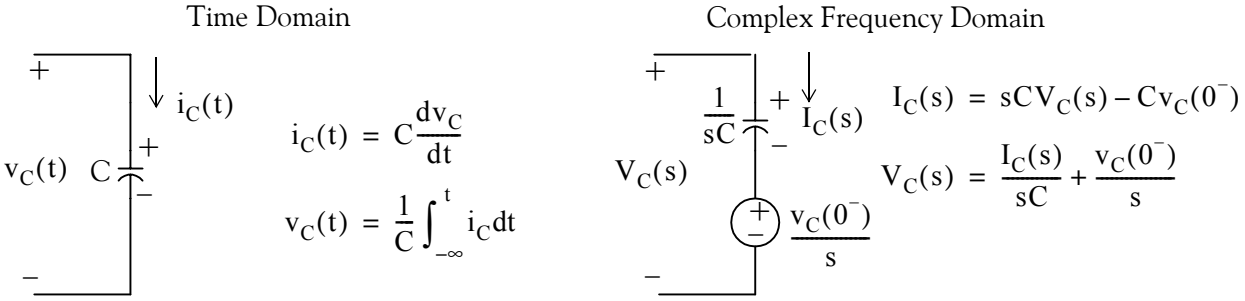


Figure 4.3. Capacitive circuit in time domain and complex frequency domain

Note:
In the complex frequency domain, the terms sL and 1/sC are referred to as *complex inductive impedance*, and *complex capacitive impedance* respectively. Likewise, the terms and sC and 1/sL are called *complex capacitive admittance* and *complex inductive admittance* respectively.

Example 4.1

Use the Laplace transform method and apply Kirchoff's Current Law (KCL) to find the voltage $v_C(t)$ across the capacitor for the circuit of Figure 4.4, given that $v_C(0^-) = 6$ V.

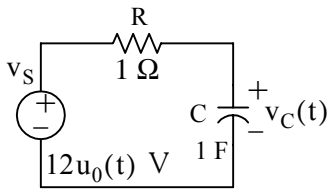


Figure 4.4. Circuit for Example 4.1

Solution:

We apply KCL at node A as shown in Figure 4.5.

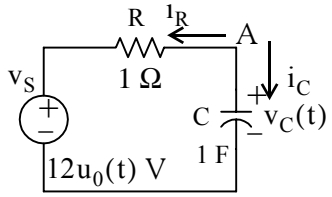


Figure 4.5. Application of KCL for the circuit of Example 4.1

Then,

$$i_R + i_C = 0$$

or

$$\frac{v_C(t) - 12u_0(t)}{1} + 1 \cdot \frac{dv_C}{dt} = 0$$

$$\frac{dv_C}{dt} + v_C(t) = 12u_0(t) \tag{4.1}$$

The Laplace transform of (4.1) is

$$sV_C(s) - v_C(0^-) + V_C(s) = \frac{12}{s}$$

$$(s+1)V_C(s) = \frac{12}{s} + 6$$

$$V_C(s) = \frac{6s+12}{s(s+1)}$$

By partial fraction expansion,

$$V_C(s) = \frac{6s+12}{s(s+1)} = \frac{r_1}{s} + \frac{r_2}{(s+1)}$$

$$r_1 = \left.\frac{6s+12}{(s+1)}\right|_{s=0} = 12$$

$$r_2 = \left.\frac{6s+12}{s}\right|_{s=-1} = -6$$

Therefore,

$$V_C(s) = \frac{12}{s} - \frac{6}{s+1} \Leftrightarrow 12 - 6e^{-t} = (12 - 6e^{-t})u_0(t) = v_C(t)$$

Example 4.2

Use the Laplace transform method and apply Kirchoff's Voltage Law (KVL) to find the voltage $v_C(t)$ across the capacitor for the circuit of Figure 4.6, given that $v_C(0^-) = 6$ V.

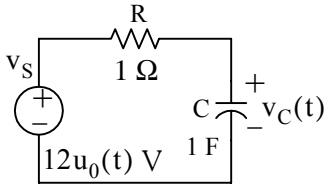


Figure 4.6. Circuit for Example 4.2

Solution:

This is the same circuit as in Example 4.1. We apply KVL for the loop shown in Figure 4.7.

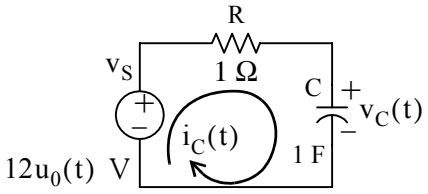


Figure 4.7. Application of KVL for the circuit of Example 4.2

$$Ri_C(t) + \frac{1}{C}\int_{-\infty}^{t} i_C(t)dt = 12u_0(t)$$

and with $R = 1$ and $C = 1$, we obtain

$$i_C(t) + \int_{-\infty}^{t} i_C(t)dt = 12u_0(t) \tag{4.2}$$

Next, taking the Laplace transform of both sides of (4.2), we obtain

$$I_C(s) + \frac{I_C(s)}{s} + \frac{v_C(0^-)}{s} = \frac{12}{s}$$

$$\left(1 + \frac{1}{s}\right)I_C(s) = \frac{12}{s} - \frac{6}{s} = \frac{6}{s}$$

$$\left(\frac{s+1}{s}\right)I_C(s) = \frac{6}{s}$$

or

$$I_C(s) = \frac{6}{s+1} \Leftrightarrow i_C(t) = 6e^{-t}u_0(t)$$

Check: From Example 4.1,

$$v_C(t) = (12 - 6e^{-t})u_0(t)$$

Then,

$$i_C(t) = C\frac{dv_C}{dt} = \frac{dv_C}{dt} = \frac{d}{dt}(12 - 6e^{-t})u_0(t) = 6e^{-t}u_0(t) + 6\delta(t) \tag{4.3}$$

The presence of the delta function in (4.3) is a result of the unit step that is applied at $t = 0$.

Example 4.3

In the circuit of Figure 4.8, switch S_1 closes at $t = 0$, while at the same time, switch S_2 opens. Use the Laplace transform method to find $v_{out}(t)$ for $t > 0$.

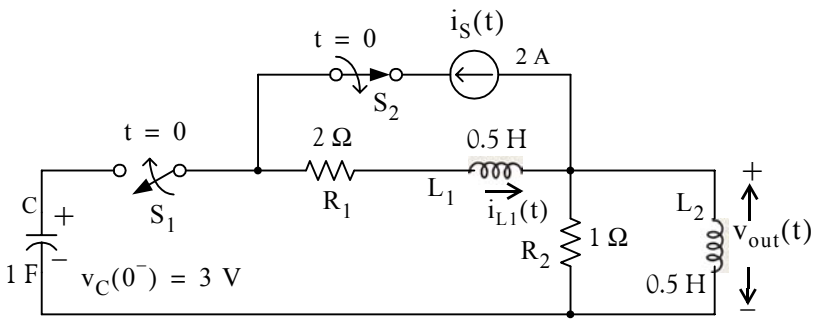


Figure 4.8. Circuit for Example 4.3

Solution:

Since the circuit contains a capacitor and an inductor, we must consider two initial conditions One is given as $v_C(0^-) = 3$ V. The other initial condition is obtained by observing that there is an initial current of 2 A in inductor L_1; this is provided by the 2 A current source just before switch S_2 opens. Therefore, our second initial condition is $i_{L1}(0^-) = 2$ A.

For $t > 0$, we transform the circuit of Figure 4.8 into its *s–domain** equivalent shown in Figure 4.9.

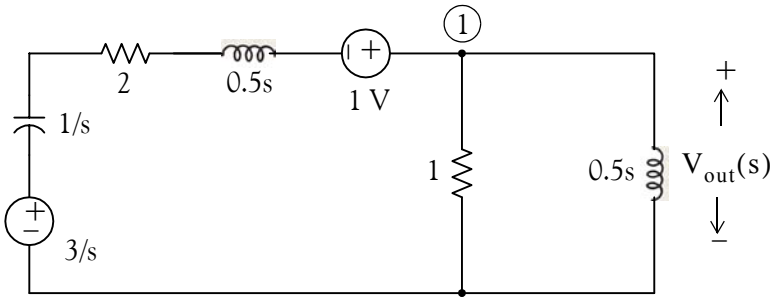


Figure 4.9. Transformed circuit of Example 4.3

In Figure 4.9 the current in inductor L_1 has been replaced by a voltage source of 1 V. This is found from the relation

$$L_1 i_{L1}(0^-) = \frac{1}{2} \times 2 = 1 \text{ V} \tag{4.4}$$

The polarity of this voltage source is as shown in Figure 4.9 so that it is consistent with the direction of the current $i_{L1}(t)$ in the circuit of Figure 4.8 just before switch S_2 opens. The initial capacitor voltage is replaced by a voltage source equal to $3/s$.

* *Henceforth, for convenience, we will refer the time domain as t–domain and the complex frequency domain as s–domain.*

Applying KCL at node ① we obtain

$$\frac{V_{out}(s) - 1 - 3/s}{1/s + 2 + s/2} + \frac{V_{out}(s)}{1} + \frac{V_{out}(s)}{s/2} = 0 \tag{4.5}$$

and after simplification

$$V_{out}(s) = \frac{2s(s+3)}{s^3 + 8s^2 + 10s + 4} \tag{4.6}$$

We will use MATLAB to factor the denominator D(s) of (4.6) into a linear and a quadratic factor.

```
p=[1  8  10  4]; r=roots(p)                          % Find the roots of D(s)

r =
  -6.5708
  -0.7146 + 0.3132i
  -0.7146 - 0.3132i

y=expand((s + 0.7146 – 0.3132j)*(s + 0.7146 + 0.3132j))   % Find quadratic form

y =
s^2+3573/2500*s+3043737/5000000

3573/2500                                            % Simplify coefficient of s

ans =
    1.4292

3043737/5000000                                      % Simplify constant term

ans =
    0.6087
```

Therefore,

$$V_{out}(s) = \frac{2s(s+3)}{s^3 + 8s^2 + 10s + 4} = \frac{2s(s+3)}{(s+6.57)(s^2 + 1.43s + 0.61)} \tag{4.7}$$

Next, we perform partial fraction expansion.

$$V_{out}(s) = \frac{2s(s+3)}{(s+6.57)(s^2 + 1.43s + 0.61)} = \frac{r_1}{s+6.57} + \frac{r_2 s + r_3}{s^2 + 1.43s + 0.61} \tag{4.8}$$

$$r_1 = \left.\frac{2s(s+3)}{s^2 + 1.43s + 0.61}\right|_{s=-6.57} = 1.36 \tag{4.9}$$

The residues r_2 and r_3 are found from the equality

$$2s(s+3) = r_1(s^2 + 1.43s + 0.61) + (r_2 s + r_3)(s + 6.57) \tag{4.10}$$

Equating constant terms of (4.10), we obtain

$$0 = 0.61r_1 + 6.57r_3$$

and by substitution of the known value of r_1 from (4.9), we obtain

$$r_3 = -0.12$$

Similarly, equating coefficients of s^2, we obtain

$$2 = r_1 + r_2$$

and using the known value of r_1, we obtain

$$r_2 = 0.64 \tag{4.11}$$

By substitution into (4.8),

$$V_{out}(s) = \frac{1.36}{s+6.57} + \frac{0.64s - 0.12}{s^2 + 1.43s + 0.61} = \frac{1.36}{s+6.57} + \frac{0.64s + 0.46 - 0.58}{s^2 + 1.43s + 0.51 + 0.1} *$$

or

$$\begin{aligned} V_{out}(s) &= \frac{1.36}{s+6.57} + (0.64)\frac{s + 0.715 - 0.91}{(s+0.715)^2 + (0.316)^2} \\ &= \frac{1.36}{s+6.57} + \frac{0.64(s+0.715)}{(s+0.715)^2 + (0.316)^2} - \frac{0.58}{(s+0.715)^2 + (0.316)^2} \\ &= \frac{1.36}{s+6.57} + \frac{0.64(s+0.715)}{(s+0.715)^2 + (0.316)^2} - \frac{1.84 \times 0.316}{(s+0.715)^2 + (0.316)^2} \end{aligned} \tag{4.12}$$

Taking the Inverse Laplace of (4.12), we obtain

$$v_{out}(t) = (1.36e^{-6.57t} + 0.64e^{-0.715t}\cos 0.316t - 1.84e^{-0.715t}\sin 0.316t)u_0(t) \tag{4.13}$$

From (4.13), we observe that as $t \to \infty$, $v_{out}(t) \to 0$. This is to be expected because $v_{out}(t)$ is the voltage across the inductor as we can see from the circuit of Figure 4.9. The MATLAB script below will plot the relation (4.13) above.

---

* *We perform these steps to express the term* $\frac{0.64s - 0.12}{s^2 + 1.43s + 0.61}$ *in a form that resembles the transform pairs* $e^{-at}\cos\omega t u_0(t) \Leftrightarrow \frac{s+a}{(s+a)^2 + \omega^2}$ *and* $e^{-at}\sin\omega t u_0(t) \Leftrightarrow \frac{\omega}{(s+a)^2 + \omega^2}$. *The remaining steps are carried out in (4.12).*

```
t=0:0.01:10;...
Vout=1.36.*exp(−6.57.*t)+0.64.*exp(−0.715.*t).*cos(0.316.*t)−1.84.*exp(−0.715.*t).*sin(0.316.*t);...
plot(t,Vout); grid
```

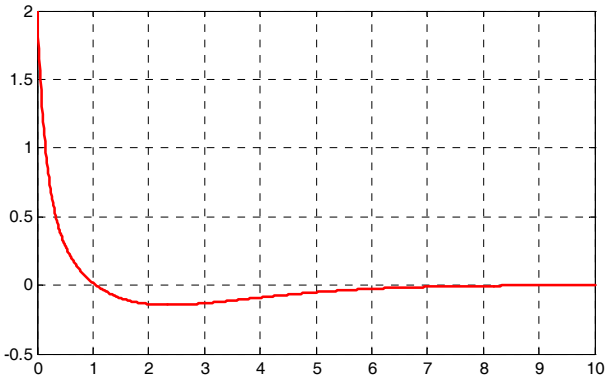


Figure 4.10. Plot of $v_{out}(t)$ for the circuit of Example 4.3

4.2 Complex Impedance Z(s)

Consider the s – domain RLC series circuit of Figure 4.11, where the initial conditions are assumed to be zero.

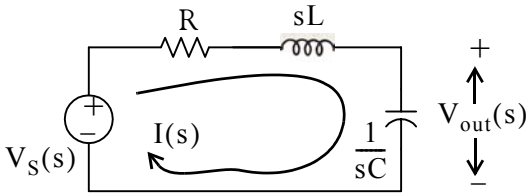


Figure 4.11. Series RLC circuit in s–domain

For this circuit, the sum $R + sL + \frac{1}{sC}$ represents the total opposition to current flow. Then,

$$I(s) = \frac{V_S(s)}{R + sL + 1/sC} \tag{4.14}$$

and defining the ratio $V_s(s)/I(s)$ as $Z(s)$, we obtain

$$\boxed{Z(s) \equiv \frac{V_S(s)}{I(s)} = R + sL + \frac{1}{sC}} \tag{4.15}$$

and thus, the s – domain current I(s) can be found from the relation (4.16) below.

$$I(s) = \frac{V_S(s)}{Z(s)} \tag{4.16}$$

where

$$Z(s) = R + sL + \frac{1}{sC} \tag{4.17}$$

We recall that $s = \sigma + j\omega$. Therefore, Z(s) is a complex quantity, and it is referred to as the *complex input impedance* of an s – domain RLC series circuit. In other words, Z(s) is the ratio of the voltage excitation $V_s(s)$ to the current response I(s) under *zero state* (zero initial conditions).

Example 4.4

For the network of Figure 4.12, all values are in Ω (ohms). Find Z(s) using:

a. nodal analysis

b. successive combinations of series and parallel impedances

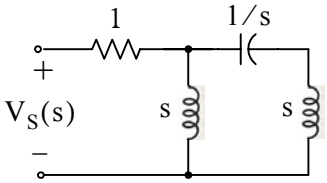


Figure 4.12. Circuit for Example 4.4

Solution:

a.

We will first find I(s), and we will compute Z(s) using (4.15). We assign the voltage $V_A(s)$ at node A as shown in Figure 4.13.

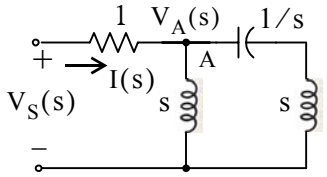


Figure 4.13. Network for finding I(s) in Example 4.4

By nodal analysis,

$$\frac{V_A(s)-V_S(s)}{1}+\frac{V_A(s)}{s}+\frac{V_A(s)}{s+1/s}=0$$

$$\left(1+\frac{1}{s}+\frac{1}{s+1/s}\right)V_A(s)=V_S(s)$$

$$V_A(s)=\frac{s^3+1}{s^3+2s^2+s+1}\cdot V_S(s)$$

The current $I(s)$ is now found as

$$I(s)=\frac{V_S(s)-V_A(s)}{1}=\left(1-\frac{s^3+1}{s^3+2s^2+s+1}\right)V_S(s)=\frac{2s^2+1}{s^3+2s^2+s+1}\cdot V_S(s)$$

and thus,

$$Z(s)=\frac{V_S(s)}{I(s)}=\frac{s^3+2s^2+s+1}{2s^2+1} \tag{4.18}$$

b.

The impedance $Z(s)$ can also be found by successive combinations of series and parallel impedances, as it is done with series and parallel resistances. For convenience, we denote the network devices as Z_1, Z_2, Z_3 and Z_4 shown in Figure 4.14.

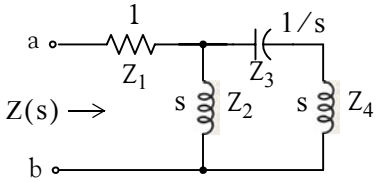


Figure 4.14. Computation of the impedance of Example 4.4 by series – parallel combinations

To find the equivalent impedance $Z(s)$, looking to the right of terminals a and b, we start on the right side of the network and we proceed to the left combining impedances as we would combine resistances where the symbol $\|$ denotes parallel combination. Then,

$$Z(s)=[(Z_3+Z_4)\,\|\,Z_2]+Z_1$$

$$Z(s)=\frac{s(s+1/s)}{s+s+1/s}+1=\frac{s^2+1}{(2s^2+1)/s}+1=\frac{s^3+s}{2s^2+1}+1=\frac{s^3+2s^2+s+1}{2s^2+1} \tag{4.19}$$

We observe that (4.19) is the same as (4.18).

4.3 Complex Admittance Y(s)

Consider the s – domain GLC parallel circuit of Figure 4.15 where the initial conditions are zero.

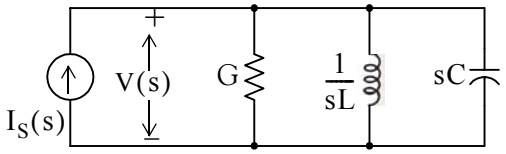


Figure 4.15. Parallel GLC circuit in s–domain

For the circuit of Figure 4.15,

$$GV(s) + \frac{1}{sL}V(s) + sCV(s) = I(s)$$

$$\left(G + \frac{1}{sL} + sC\right)(V(s)) = I(s)$$

Defining the ratio $I_S(s)/V(s)$ as $Y(s)$, we obtain

$$Y(s) \equiv \frac{I(s)}{V(s)} = G + \frac{1}{sL} + sC = \frac{1}{Z(s)} \tag{4.20}$$

and thus the s – domain voltage $V(s)$ can be found from

$$\boxed{V(s) = \frac{I_S(s)}{Y(s)}} \tag{4.21}$$

where

$$\boxed{Y(s) = G + \frac{1}{sL} + sC} \tag{4.22}$$

We recall that $s = \sigma + j\omega$. Therefore, $Y(s)$ is a complex quantity, and it is referred to as the *complex input admittance* of an s – domain GLC parallel circuit. In other words, $Y(s)$ is the ratio of the current excitation $I_S(s)$ to the voltage response $V(s)$ under *zero state* (zero initial conditions).

Example 4.5

Compute $Z(s)$ and $Y(s)$ for the circuit of Figure 4.16. All values are in Ω (ohms). Verify your answers with MATLAB.

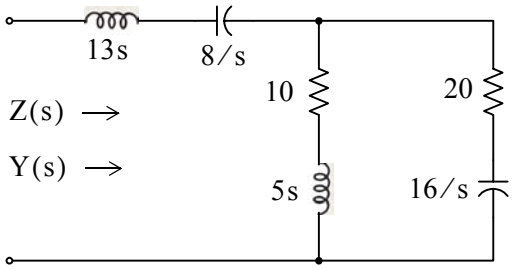


Figure 4.16. Circuit for Example 4.5

Solution:

It is convenient to represent the given circuit as shown in Figure 4.17.

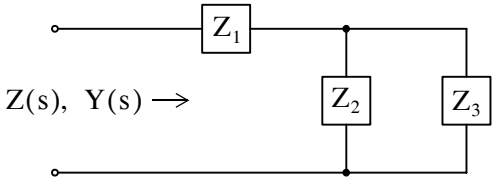


Figure 4.17. Simplified circuit for Example 4.5

where

$$Z_1 = 13s + \frac{8}{s} = \frac{13s^2 + 8}{s}$$

$$Z_2 = 10 + 5s$$

$$Z_3 = 20 + \frac{16}{s} = \frac{4(5s + 4)}{s}$$

Then,

$$Z(s) = Z_1 + \frac{Z_2 Z_3}{Z_2 + Z_3} = \frac{13s^2 + 8}{s} + \frac{(10 + 5s)\left(\frac{4(5s + 4)}{s}\right)}{10 + 5s + \frac{4(5s + 4)}{s}} = \frac{13s^2 + 8}{s} + \frac{(10 + 5s)\left(\frac{4(5s + 4)}{s}\right)}{\frac{5s^2 + 10s + 4(5s + 4)}{s}}$$

$$= \frac{13s^2 + 8}{s} + \frac{20(5s^2 + 14s + 8)}{5s^2 + 30s + 16} = \frac{65s^4 + 490s^3 + 528s^2 + 400s + 128}{s(5s^2 + 30s + 16)}$$

Check with MATLAB:

```
syms s;                % Define symbolic variable s
z1 = 13*s + 8/s; z2 = 5*s + 10; z3 = 20 + 16/s; z = z1 + z2 * z3 / (z2+z3)

z =
13*s+8/s+(5*s+10)*(20+16/s)/(5*s+30+16/s)
```

```
z10 = simplify(z)

z10 =
(65*s^4+490*s^3+528*s^2+400*s+128)/s/(5*s^2+30*s+16)

pretty(z10)

     4         3         2
65 s  + 490 s  + 528 s  + 400 s + 128
-------------------------------------
                2
        s (5 s  + 30 s + 16)
```

The complex input admittance $Y(s)$ is found by taking the reciprocal of $Z(s)$, that is,

$$Y(s) = \frac{1}{Z(s)} = \frac{s(5s^2 + 30s + 16)}{65s^4 + 490s^3 + 528s^2 + 400s + 128} \tag{4.23}$$

4.4 Transfer Functions

In an $s-$domain circuit, the ratio of the output voltage $V_{out}(s)$ to the input voltage $V_{in}(s)$ *under zero state conditions*, is of great interest* in network analysis. This ratio is referred to as the *voltage transfer function* and it is denoted as $G_v(s)$, that is,

$$G_v(s) \equiv \frac{V_{out}(s)}{V_{in}(s)} \tag{4.24}$$

Similarly, the ratio of the output current $I_{out}(s)$ to the input current $I_{in}(s)$ *under zero state conditions*, is called the *current transfer function* denoted as $G_i(s)$, that is,

$$G_i(s) \equiv \frac{I_{out}(s)}{I_{in}(s)} \tag{4.25}$$

The current transfer function of (4.25) is rarely used; therefore, from now on, the transfer function will have the meaning of the voltage transfer function, i.e.,

* *To appreciate the usefulness of the transfer function, let us express relation (4.24) as* $V_{out}(s) = G_v(s) \cdot V_{in}(s)$. *This relation indicates that if we know the transfer function of a network, we can compute its output by multiplication of the transfer function by its input. We should also remember that the transfer function concept exists only in the complex frequency domain. In the time domain this concept is known as the* ***impulse response****, and it is discussed in Chapter 6 of this text.*

$$G(s) \equiv \frac{V_{out}(s)}{V_{in}(s)} \tag{4.26}$$

Example 4.6

Derive an expression for the transfer function $G(s)$ for the circuit of Figure 4.18, where R_g represents the internal resistance of the applied (source) voltage V_S, and R_L represents the resistance of the load that consists of R_L, L, and C.

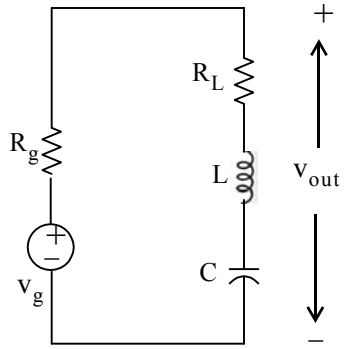


Figure 4.18. Circuit for Example 4.6

Solution:

No initial conditions are given, and even if they were, we would disregard them since the transfer function was defined as the ratio of the output voltage $V_{out}(s)$ to the input voltage $V_{in}(s)$ under zero initial conditions. The $s-domain$ circuit is shown in Figure 4.19.

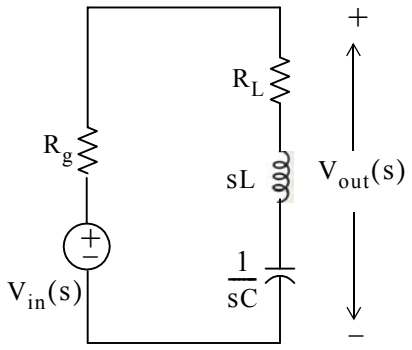


Figure 4.19. The s–domain circuit for Example 4.6

The transfer function $G(s)$ is readily found by application of the voltage division expression of the $s-domain$ circuit of Figure 4.19. Thus,

$$V_{out}(s) = \frac{R_L + sL + 1/sC}{R_g + R_L + sL + 1/sC} V_{in}(s)$$

Therefore,

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{R_L + Ls + 1/sC}{R_g + R_L + Ls + 1/sC} \tag{4.27}$$

Example 4.7

Compute the transfer function $G(s)$ for the circuit of Figure 4.20 in terms of the circuit constants R_1, R_2, R_3, C_1, and C_2 Then, replace the complex variable s with $j\omega$, and the circuit constants with their numerical values and plot the magnitude $|G(s)| = V_{out}(s)/V_{in}(s)$ versus radian frequency ω.

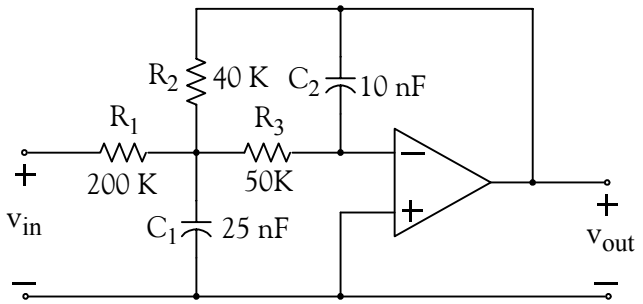


Figure 4.20. Circuit for Example 4.7

Solution:

The complex frequency domain equivalent circuit is shown in Figure 4.21.

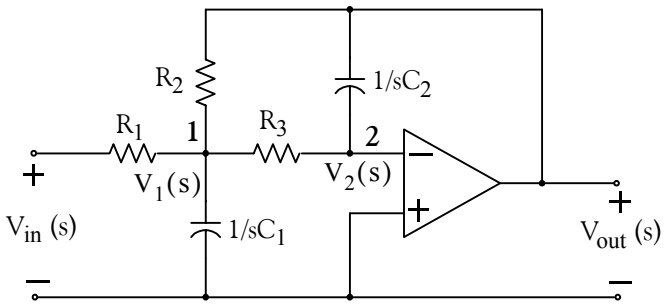


Figure 4.21. The s–domain circuit for Example 4.7

Next, we write nodal equations at nodes **1** and **2**. At node **1**,

$$\frac{V_1(s) - V_{in}(s)}{R_1} + \frac{V_1}{1/sC_1} + \frac{V_1(s) - V_{out}(s)}{R_2} + \frac{V_1(s) - V_2(s)}{R_3} = 0 \tag{4.28}$$

At node 2,

$$\frac{V_2(s)-V_1(s)}{R_3} = \frac{V_{out}(s)}{1/sC_2} \tag{4.29}$$

Since $V_2(s) = 0$ (virtual ground), we express (4.29) as

$$V_1(s) = (-sR_3C_2)V_{out}(s) \tag{4.30}$$

and by substitution of (4.30) into (4.28), rearranging, and collecting like terms, we obtain:

$$\left[\left(\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}+sC_1\right)(-sR_3C_2)-\frac{1}{R_2}\right]V_{out}(s) = \frac{1}{R_1}V_{in}(s)$$

or

$$G(s)=\frac{V_{out}(s)}{V_{in}(s)} = \frac{-1}{R_1\left[\left(1/R_1+1/R_2+1/R_3+sC_1\right)(sR_3C_2)+1/R_2\right]} \tag{4.31}$$

To simplify the denominator of (4.31), we use the MATLAB script below with the given values of the resistors and the capacitors.

```
syms s;                 % Define symbolic variable s
R1=2*10^5; R2=4*10^4; R3=5*10^4; C1=25*10^(-9); C2=10*10^(-9);...
DEN=R1*((1/R1+1/R2+1/R3+s*C1)*(s*R3*C2)+1/R2); simplify(DEN)

ans =
1/200*s+18889465931478582 5/7555786372591432341 9136*s^2+5

18889465931478582 5/7555786372591432341 9136    % Simplify coefficient of s^2

ans =
  2.5000e-006

1/200                                           % Simplify coefficient of s^2

ans =
    0.0050
```

Therefore,

$$G(s)=\frac{V_{out}(s)}{V_{in}(s)} = \frac{-1}{2.5\times10^{-6}s^2+5\times10^{-3}s+5}$$

By substitution of s with $j\omega$ we obtain

$$G(j\omega)=\frac{V_{out}(j\omega)}{V_{in}(j\omega)} = \frac{-1}{2.5\times10^{-6}\omega^2-j5\times10^{-3}\omega+5} \tag{4.32}$$

We use MATLAB to plot the magnitude of (4.32) on a semilog scale with the following script:

```
w=1:10:10000; Gs=–1./(2.5.*10.^(–6).*w.^2–5.*j.*10.^(–3).*w+5);...
semilogx(w,abs(Gs)); xlabel('Radian Frequency w'); ylabel('|Vout/Vin|');...
title('Magnitude Vout/Vin vs. Radian Frequency'); grid
```

The plot is shown in Figure 4.22. We observe that the given op amp circuit is a second order low–pass filter whose cutoff frequency (–3 dB) occurs at about 700 r/s.

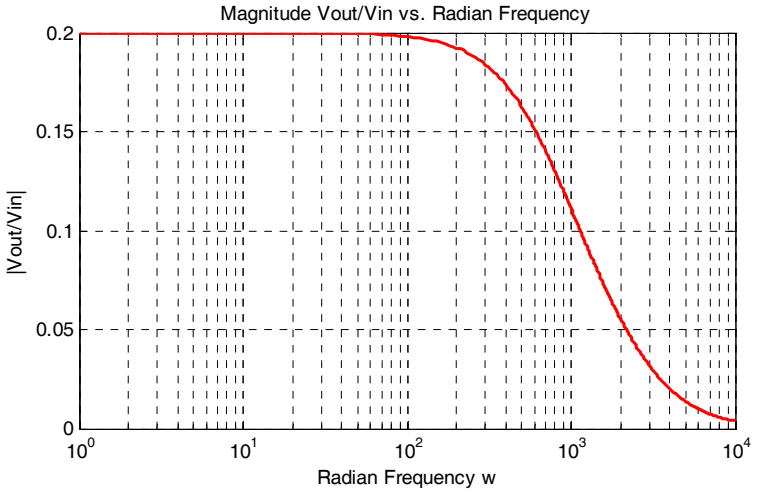


Figure 4.22. $|G(j\omega)|$ versus ω for the circuit of Example 4.7

4.5 Using the Simulink Transfer Fcn Block

The Simulink **Transfer Fcn** block implements a transfer function where the input $V_{IN}(s)$ and the output $V_{OUT}(s)$ can be expressed in transfer function form as

$$G(s) = \frac{V_{OUT}(s)}{V_{IN}(s)} \tag{4.33}$$

Example 4.8

Let us reconsider the active low–pass filter op amp circuit of Figure 4.21, Page 4-15 where we found that the transfer function is

$$G(s)=\frac{V_{out}(s)}{V_{in}(s)}=\frac{-1}{R_1\left[\left(1/R_1+1/R_2+1/R_3+sC_1\right)(sR_3C_2)+1/R_2\right]} \tag{4.34}$$

and for simplicity, let $R_1 = R_2 = R_3 = 1\ \Omega$, and $C_1 = C_2 = 1\ \text{F}$. By substitution into (4.34) we obtain

$$G(s)=\frac{V_{out}(s)}{V_{in}(s)}=\frac{-1}{s^2+3s+1} \tag{4.35}$$

Next, we let the input be the unit step function $u_0(t)$, and as we know from Chapter 2, $u_0(t) \Leftrightarrow 1/s$. Therefore,

$$V_{out}(s) = G(s)\cdot V_{in}(s) = \frac{1}{s}\cdot\frac{-1}{s^2+3s+1} = \frac{-1}{s^3+3s^2+s} \tag{4.36}$$

To find $v_{out}(t)$, we perform partial fraction expansion, and for convenience, we use the MATLAB **residue** function as follows:

```
num=–1; den=[1 3 1 0];[r p k]=residue(num,den)
```

```
r =
   -0.1708
    1.1708
   -1.0000
p =
   -2.6180
   -0.3820
         0
k =
     []
```

Therefore,

$$\left(\frac{1}{s}\cdot\frac{-1}{s^2+3s+1}=-\frac{1}{s}+\frac{1.171}{s+0.382}-\frac{0.171}{s+2.618}\right)\Leftrightarrow -1+1.171e^{-0.382t}-0.171e^{-2.618t} = v_{out}(t) \tag{4.37}$$

The plot for $v_{out}(t)$ is obtained with the following MATLAB script, and it is shown in Figure 4.23.

```
t=0:0.01:10; ft=–1+1.171.*exp(–0.382.*t)–0.171.*exp(–2.618.*t); plot(t,ft); grid
```

The same plot can be obtained using the Simulink model of Figure 4.24, where in the Function Block Parameters dialog box for the Transfer Fcn block we enter –1 for the numerator, and [1 3 1] for the denominator. After the simulation command is executed, the Scope block displays the waveform of Figure 4.25.

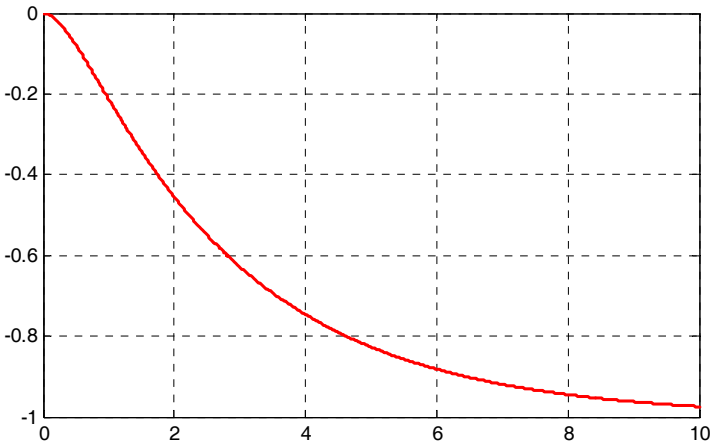


Figure 4.23. Plot of $v_{out}(t)$ for Example 4.8.

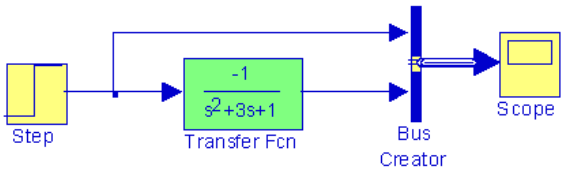


Figure 4.24. Simulink model for Example 4.8

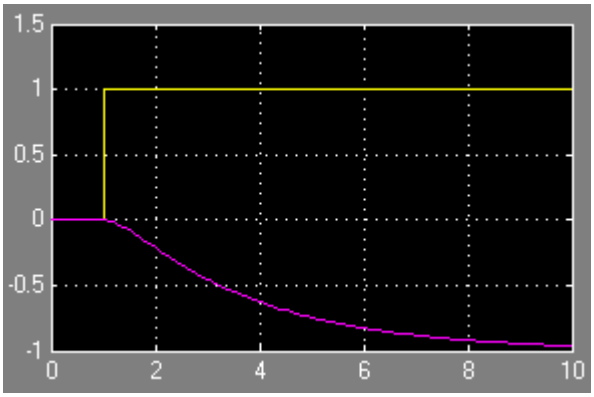


Figure 4.25. Waveform for the Simulink model of Figure 4.24

4.6 Summary

- The Laplace transformation provides a convenient method of analyzing electric circuits since integrodifferential equations in the $t-\text{domain}$ are transformed to algebraic equations in the $s-\text{domain}$.
- In the $s-\text{domain}$ the terms sL and $1/sC$ are called complex inductive impedance, and complex capacitive impedance respectively. Likewise, the terms and sC and $1/sL$ are called complex capacitive admittance and complex inductive admittance respectively.
- The expression

$$Z(s) = R + sL + \frac{1}{sC}$$

  is a complex quantity, and it is referred to as the complex input impedance of an $s-\text{domain}$ RLC series circuit.
- In the $s-\text{domain}$ the current $I(s)$ can be found from

$$I(s) = \frac{V_S(s)}{Z(s)}$$

- The expression

$$Y(s) = G + \frac{1}{sL} + sC$$

  is a complex quantity, and it is referred to as the complex input admittance of an $s-\text{domain}$ GLC parallel circuit.
- In the $s-\text{domain}$ the voltage $V(s)$ can be found from

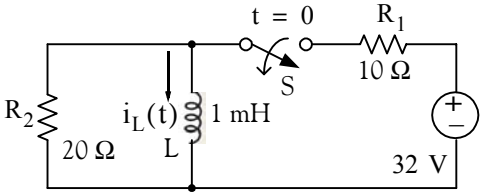
$$V(s) = \frac{I_S(s)}{Y(s)}$$

- In an $s-\text{domain}$ circuit, the ratio of the output voltage $V_{out}(s)$ to the input voltage $V_{in}(s)$ under zero state conditions is referred to as the voltage transfer function and it is denoted as $G(s)$, that is,

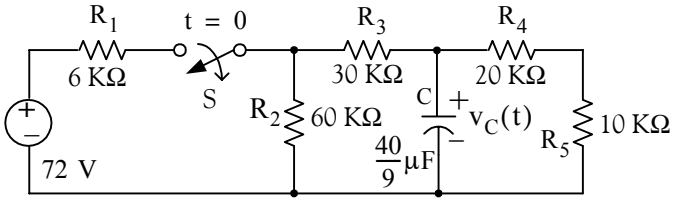
$$G(s) \equiv \frac{V_{out}(s)}{V_{in}(s)}$$

4.7 Exercises

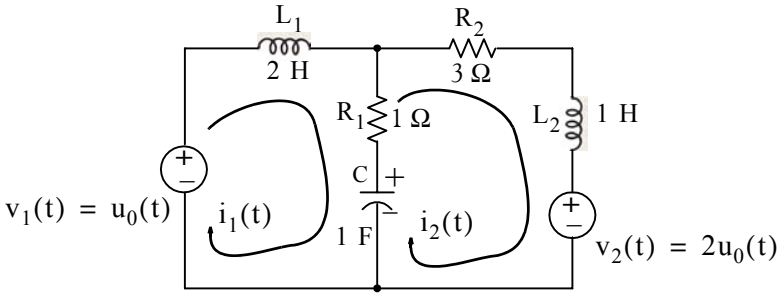
1. In the circuit below, switch S has been closed for a long time, and opens at $t = 0$. Use the Laplace transform method to compute $i_L(t)$ for $t > 0$.



2. In the circuit below, switch S has been closed for a long time, and opens at $t = 0$. Use the Laplace transform method to compute $v_c(t)$ for $t > 0$.



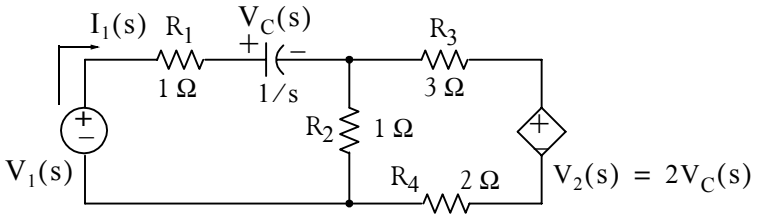
3. Use mesh analysis and the Laplace transform method, to compute $i_1(t)$ and $i_2(t)$ for the circuit below, given that $i_L(0^-) = 0$ and $v_C(0^-) = 0$.



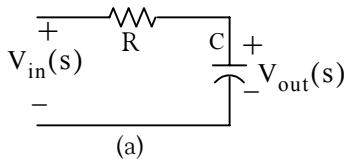
4. For the s – domain circuit below,

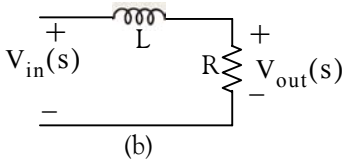
a. compute the admittance $Y(s) = I_1(s)/V_1(s)$

b. compute the t – domain value of $i_1(t)$ when $v_1(t) = u_0(t)$, and all initial conditions are zero.

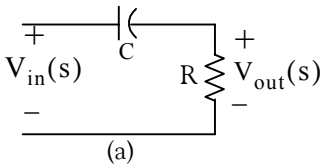


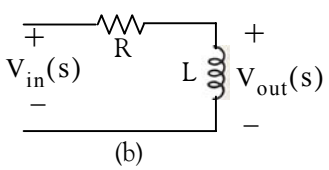
5. Derive the transfer functions for the networks (a) and (b) below.



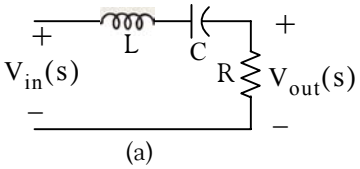


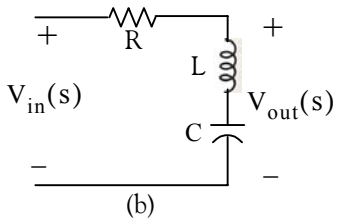
6. Derive the transfer functions for the networks (a) and (b) below.



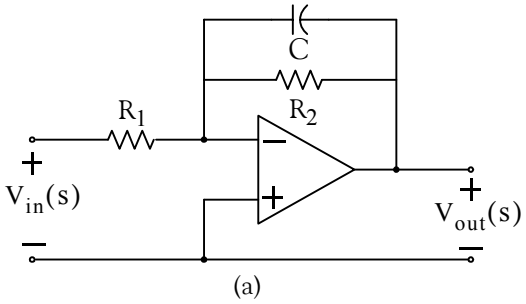


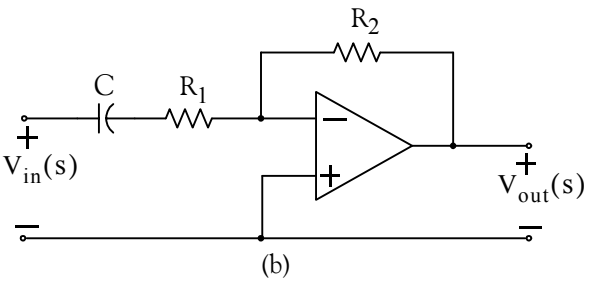
7. Derive the transfer functions for the networks (a) and (b) below.



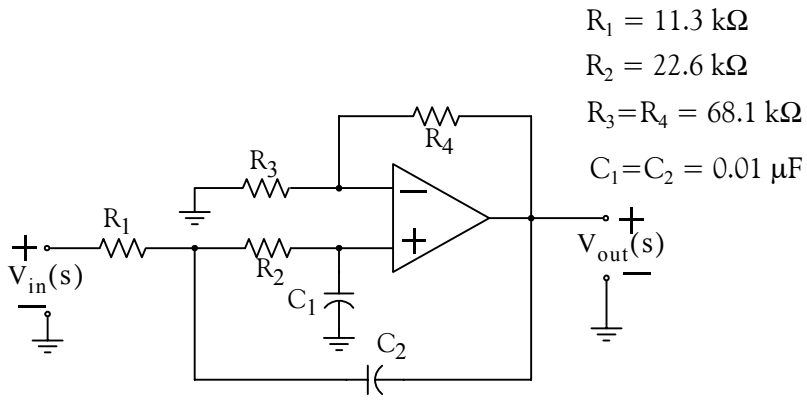


8. Derive the transfer function for the networks (a) and (b) below.



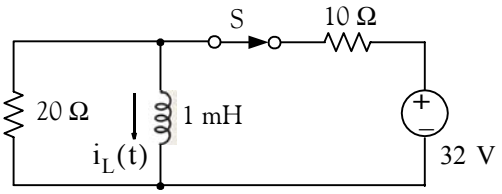


9. Derive the transfer function for the network below. Using MATLAB, plot $|G(s)|$ versus frequency in Hertz, on a semilog scale.



4.8 Solutions to End–of–Chapter Exercises

1. At $t = 0^-$, the switch is closed, and the $t-\text{domain}$ circuit is as shown below where the 20 Ω resistor is shorted out by the inductor.

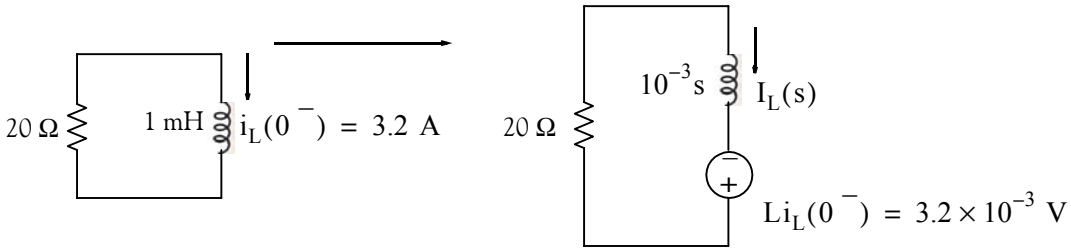


Then,

$$i_L(t)\Big|_{t=0^-} = \frac{32}{10} = 3.2 \text{ A}$$

and thus the initial condition has been established as $i_L(0^-) = 3.2$ A

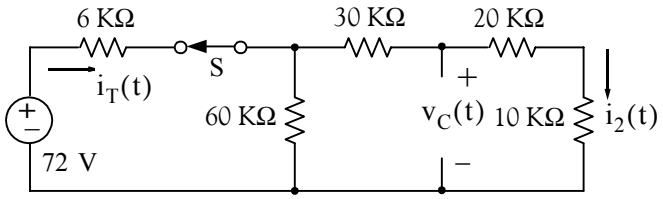
For all $t > 0$ the $t-\text{domain}$ and $s-\text{domain}$ circuits are as shown below.



From the $s-\text{domain}$ circuit on the right side above we obtain

$$I_L(s) = \frac{3.2 \times 10^{-3}}{20 + 10^{-3}s} = \frac{3.2}{s + 20000} \Leftrightarrow 3.2e^{-20000t}u_0(t) = i_L(t)$$

2. At $t = 0^-$, the switch is closed and the $t-\text{domain}$ circuit is as shown below.



Then,

$$i_T(0^-) = \frac{72\text{ V}}{6\text{ K}\Omega + 60\text{ K}\Omega \,\|\, 60\text{ K}\Omega} = \frac{72\text{ V}}{6\text{ K}\Omega + 30\text{ K}\Omega} = \frac{72\text{ V}}{36\text{ K}\Omega} = 2\text{ mA}$$

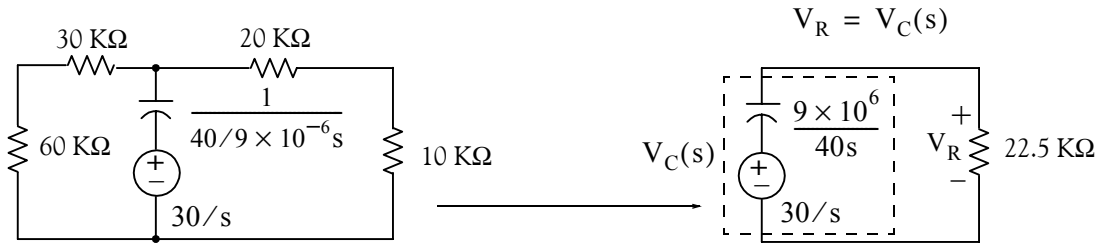
and

$$i_2(0^-) = \frac{1}{2} i_T(0^-) = 1\text{ mA}$$

Therefore, the initial condition is

$$v_C(0^-) = (20\text{ K}\Omega + 10\text{ K}\Omega) \cdot i_2(0^-) = (30\text{ K}\Omega) \cdot (1\text{ mA}) = 30\text{ V}$$

For all $t > 0$, the $s-$domain circuit is as shown below.



$$(60\text{ K}\Omega + 30\text{ K}\Omega) \,\|\, (20\text{ K}\Omega + 10\text{ K}\Omega) = 22.5\text{ K}\Omega$$

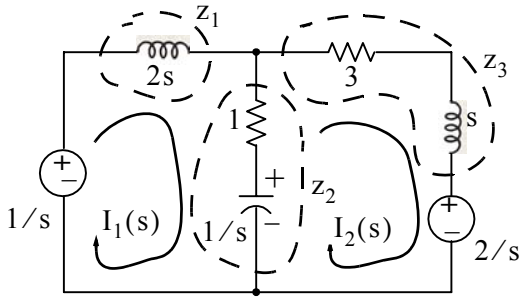
$$V_C(s) = V_R = \frac{22.5 \times 10^3}{9 \times 10^6/40s + 22.5 \times 10^3} \cdot \frac{30}{s} = \frac{30 \times 22.5 \times 10^3}{9 \times 10^6/40 + 22.5 \times 10^3 s}$$

$$= \frac{(30 \times 22.5 \times 10^3)/(22.5 \times 10^3)}{9 \times 10^6/(40 \times 22.5 \times 10^3) + s} = \frac{30}{9 \times 10^6/90 \times 10^4 + s} = \frac{30}{10 + s}$$

Then,

$$V_C(s) = \frac{30}{s + 10} \Leftrightarrow 30e^{-10t} u_0(t)\text{ V} = v_C(t)$$

3. The $s-$domain circuit is shown below where $z_1 = 2s$, $z_2 = 1 + 1/s$, and $z_3 = s + 3$



Then,

$$(z_1 + z_2)I_1(s) - z_2 I_2(s) = 1/s$$
$$-z_2 I_1(s) + (z_2 + z_3)I_2(s) = -2/s$$

and in matrix form

$$\begin{bmatrix} (z_1 + z_2) & -z_2 \\ -z_2 & (z_2 + z_3) \end{bmatrix} \cdot \begin{bmatrix} I_1(s) \\ I_2(s) \end{bmatrix} = \begin{bmatrix} 1/s \\ -2/s \end{bmatrix}$$

We use the MATLAB script below we obtain the values of the currents.

```
Z=[z1+z2  -z2; -z2  z2+z3]; Vs=[1/s  -2/s]'; Is=Z\Vs; fprintf(' \n');...
disp('Is1 = '); pretty(Is(1)); disp('Is2 = '); pretty(Is(2))
```

```
Is1 =
                                  2
                       2 s - 1 + s
  -------------------------------
                         2       3
  (6 s + 3 + 9 s  + 2 s )
Is2 =
                         2
                    4 s  + s + 1
- -------------------------------
                         2       3
   (6 s + 3 + 9 s  + 2 s ) conj(s)
```

Therefore,

$$I_1(s) = \frac{s^2 + 2s - 1}{2s^3 + 9s^2 + 6s + 3} \quad (1)$$

$$I_2(s) = -\frac{4s^2 + s + 1}{2s^3 + 9s^2 + 6s + 3} \quad (2)$$

We use MATLAB to express the denominators of (1) and (2) as a product of a linear and a quadratic term.

```
p=[2  9  6  3]; r=roots(p); fprintf(' \n'); disp('root1 ='); disp(r(1));...
disp('root2 ='); disp(r(2)); disp('root3 ='); disp(r(3)); disp('root2 + root3 ='); disp(r(2)+r(3));...
disp('root2 * root3 ='); disp(r(2)*r(3))
```

```
root1 =
   -3.8170

root2 =
  -0.3415 + 0.5257i
```

```
root3 =
 -0.3415 - 0.5257i

root2 + root3 =
    -0.6830

root2 * root3 =
     0.3930
```

and with these values (1) is written as

$$I_1(s) = \frac{s^2 + 2s - 1}{(s + 3.817) \cdot (s^2 + 0.683s + 0.393)} = \frac{r_1}{(s + 3.817)} + \frac{r_2 s + r_3}{(s^2 + 0.683s + 0.393)} \quad (3)$$

Multiplying every term by the denominator and equating numerators we obtain

$$s^2 + 2s - 1 = r_1(s^2 + 0.683s + 0.393) + (r_2 s + r_3)(s + 3.817)$$

Equating s^2, s, and constant terms we obtain

$$\begin{aligned} r_1 + r_2 &= 1 \\ 0.683r_1 + 3.817r_2 + r_3 &= 2 \\ 0.393r_1 + 3.817r_3 &= -1 \end{aligned}$$

We will use MATLAB to find these residues.

```
A=[1  1  0; 0.683  3.817  1; 0.393  0  3.817]; B=[1  2  −1]'; r=A\B; fprintf(' \n');...
fprintf('r1 = %5.2f \t',r(1)); fprintf('r2 = %5.2f \t',r(2)); fprintf('r3 = %5.2f',r(3))
```

```
r1 = 0.48    r2 = 0.52    r3 = -0.31
```

By substitution of these values into (3) we obtain

$$I_1(s) = \frac{r_1}{(s + 3.817)} + \frac{r_2 s + r_3}{(s^2 + 0.683s + 0.393)} = \frac{0.48}{(s + 3.817)} + \frac{0.52s - 0.31}{(s^2 + 0.683s + 0.393)} \quad (4)$$

By inspection, the Inverse Laplace of first term on the right side of (4) is

$$\frac{0.48}{(s + 3.82)} \Leftrightarrow 0.48e^{-3.82t} \quad (5)$$

The second term on the right side of (4) requires some manipulation. Therefore, we will use the MATLAB **ilaplace(s)** function to find the Inverse Laplace as shown below.

```
syms s t
IL=ilaplace((0.52*s-0.31)/(s^2+0.68*s+0.39));
pretty(IL)
```

```
  1217        17       1/2              1/2
- ----  exp(- -- t) 14     sin(7/50 14     t)
  4900        50
       13         17                  1/2
     + --  exp(- -- t)  cos(7/50 14     t)
       25         50
```

Thus,

$$i_1(t) = 0.48e^{-3.82t} - 0.93e^{-0.34t}\sin 0.53t + 0.52e^{-0.34t}\cos 0.53t$$

Next, we will find $I_2(s)$. We found earlier that

$$I_2(s) = -\frac{4s^2 + s + 1}{2s^3 + 9s^2 + 6s + 3}$$

and following the same procedure we obtain

$$I_2(s) = \frac{-4s^2 - s - 1}{(s + 3.817)\cdot(s^2 + 0.683s + 0.393)} = \frac{r_1}{(s + 3.817)} + \frac{r_2 s + r_3}{(s^2 + 0.683s + 0.393)} \quad (6)$$

Multiplying every term by the denominator and equating numerators we obtain

$$-4s^2 - s - 1 = r_1(s^2 + 0.683s + 0.393) + (r_2 s + r_3)(s + 3.817)$$

Equating s^2, s, and constant terms, we obtain

$$\begin{aligned} r_1 + r_2 &= -4 \\ 0.683r_1 + 3.817r_2 + r_3 &= -1 \\ 0.393r_1 + 3.817r_3 &= -1 \end{aligned}$$

We will use MATLAB to find these residues.

```
A=[1  1  0; 0.683  3.817  1; 0.393  0  3.817]; B=[-4  -1  -1]'; r=A\B; fprintf(' \n');...
fprintf('r1 = %5.2f \t',r(1)); fprintf('r2 = %5.2f \t',r(2)); fprintf('r3 = %5.2f',r(3))
```

```
r1 = -4.49    r2 = 0.49    r3 = 0.20
```

By substitution of these values into (6) we obtain

$$I_1(s) = \frac{r_1}{(s + 3.817)} + \frac{r_2 s + r_3}{(s^2 + 0.683s + 0.393)} = \frac{-4.49}{(s + 3.817)} + \frac{0.49s + 0.20}{(s^2 + 0.683s + 0.393)} \quad (7)$$

By inspection, the Inverse Laplace of first term on the right side of (7) is

$$\frac{0.48}{(s+3.82)} \Leftrightarrow -4.47e^{-3.82t} \quad (8)$$

The second term on the right side of (7) requires some manipulation. Therefore, we will use the MATLAB **ilaplace(s)** function to find the Inverse Laplace as shown below.

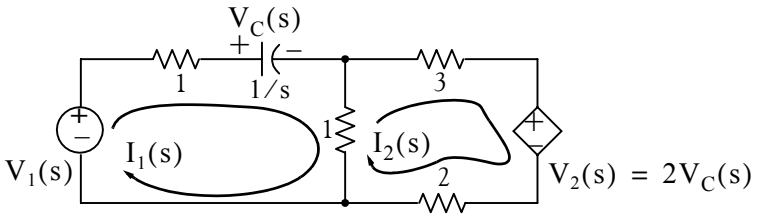
```
syms s t
IL=ilaplace((0.49*s+0.20)/(s^2+0.68*s+0.39)); pretty(IL)
```

```
 167        17     1/2             1/2
---- exp(- -- t) 14     sin(7/50 14     t)
 9800       50

   49       17                  1/2
+ --- exp(- -- t) cos(7/50 14     t)
  100       50
```

Thus,

$$i_2(t) = -4.47e^{-3.82t} + 0.06e^{-0.34t}\sin 0.53t + 0.49e^{-0.34t}\cos 0.53t$$

4.



a. Mesh 1:

$$(2 + 1/s)\cdot I_1(s) - I_2(s) = V_1(s)$$

or

$$6(2 + 1/s)\cdot I_1(s) - 6I_2(s) = 6V_1(s) \quad (1)$$

Mesh 2:

$$-I_1(s) + 6I_2(s) = -V_2(s) = -(2/s)I_1(s) \quad (2)$$

Addition of (1) and (2) yields

$$(12 + 6/s)\cdot I_1(s) + (2/s - 1)\cdot I_1(s) = 6V_1(s)$$

or

$$(11 + 8/s)\cdot I_1(s) = 6V_1(s)$$

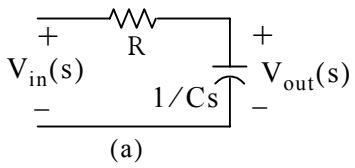
and thus

$$Y(s) = \frac{I_1(s)}{V_1(s)} = \frac{6}{11 + 8/s} = \frac{6s}{11s + 8}$$

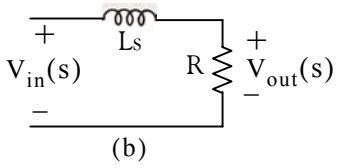
b. With $V_1(s) = 1/s$ we obtain

$$I_1(s) = Y(s) \cdot V_1(s) = \frac{6s}{11s+8} \cdot \frac{1}{s} = \frac{6}{11s+8} = \frac{6/11}{s+8/11} \Leftrightarrow \frac{6}{11} e^{-(8/11)t} = i_1(t)$$

5.



(a)



(b)

Network (a):

$$V_{out}(s) = \frac{1/Cs}{R + 1/Cs} \cdot V_{in}(s)$$

and thus

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1/Cs}{R + 1/Cs} = \frac{1/Cs}{(RCs+1)/(Cs)} = \frac{1}{RCs+1} = \frac{1/RC}{s+1/RC}$$

Network (b):

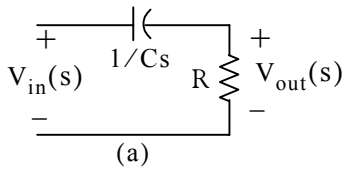
$$V_{out}(s) = \frac{R}{Ls + R} \cdot V_{in}(s)$$

and thus

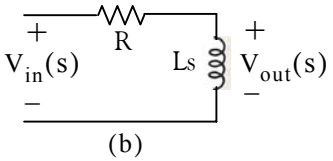
$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{R}{Ls+R} = \frac{R/L}{s+R/L}$$

Both of these networks are first–order low–pass filters.

6.



(a)



(b)

Network (a):

$$V_{out}(s) = \frac{R}{1/Cs + R} \cdot V_{in}(s)$$

and

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{R}{1/Cs+R} = \frac{RCs}{(RCs+1)} = \frac{s}{s+1/RC}$$

Network (b):

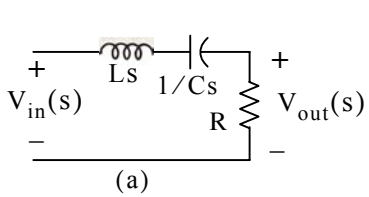
$$V_{out}(s) = \frac{Ls}{R + Ls} \cdot V_{in}(s)$$

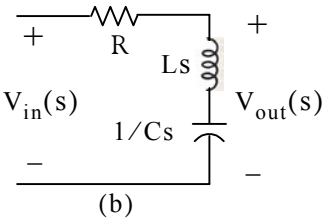
and

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Ls}{R + Ls} = \frac{s}{s + R/L}$$

Both of these networks are first–order high–pass filters.

7.





Network (a):

$$V_{out}(s) = \frac{R}{Ls + 1/Cs + R} \cdot V_{in}(s)$$

and thus

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{R}{Ls + 1/Cs + R} = \frac{RCs}{LCs^2 + 1 + RCs} = \frac{(R/L)s}{s^2 + (R/L)s + 1/LC}$$

This network is a second–order band–pass filter.

Network (b):

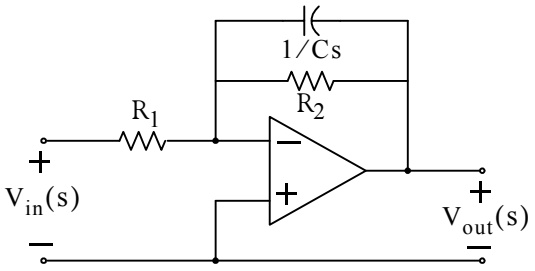
$$V_{out}(s) = \frac{Ls + 1/Cs}{R + Ls + 1/Cs} \cdot V_{in}(s)$$

and

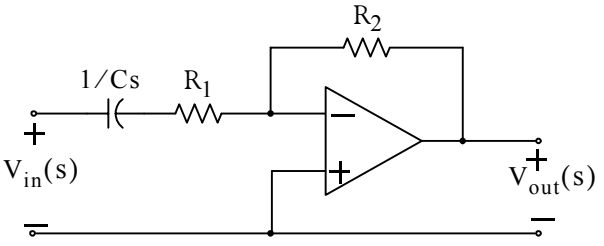
$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Ls + 1/Cs}{R + Ls + 1/Cs} = \frac{LCs^2 + 1}{LCs^2 + RCs + 1} = \frac{s^2 + 1/LC}{s^2 + (R/L)s + 1/LC}$$

This network is a second–order band–elimination (band–reject) filter.

8.



(a)



(b)

Network (a):

Let $z_1 = R_1$ and $z_2 = R_2 \| 1/Cs = \dfrac{R_2 \times 1/Cs}{R_2 + 1/Cs}$. For inverting op amps $\dfrac{V_{out}(s)}{V_{in}(s)} = -\dfrac{z_2}{z_1}$, and thus

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{-[(R_2 \times 1/Cs)/(R_2 + 1/Cs)]}{R_1} = \frac{-(R_2 \times 1/Cs)}{R_1 \cdot (R_2 + 1/Cs)} = \frac{-R_1 C}{s + 1/R_2 C}$$

This network is a first–order active low–pass filter.

Network (b):

Let $z_1 = R_1 + 1/Cs$ and $z_2 = R_2$. For inverting op-amps $\dfrac{V_{out}(s)}{V_{in}(s)} = -\dfrac{z_2}{z_1}$, and thus

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{-R_2}{R_1 + 1/Cs} = \frac{-(R_2/R_1)s}{s + 1/R_1 C}$$

This network is a first–order active high–pass filter.

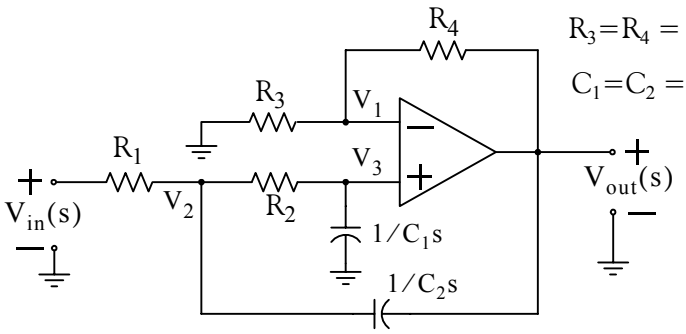
9.

$R_1 = 11.3\ K\Omega$

$R_2 = 22.6\ K\Omega$

$R_3 = R_4 = 68.1\ K\Omega$

$C_1 = C_2 = 0.01\ \mu F$



At Node V_1:

$$\frac{V_1(s)}{R_3} + \frac{V_1(s) - V_{out}(s)}{R_4} = 0$$

$$\left(\frac{1}{R_3} + \frac{1}{R_4}\right)V_1(s) = \frac{1}{R_4}V_{out}(s) \quad (1)$$

At Node V_3:

$$\frac{V_3(s) - V_2(s)}{R_2} + \frac{V_3(s)}{1/C_1 s} = 0$$

and since $V_3(s) \approx V_1(s)$, we express the last relation above as

$$\frac{V_1(s) - V_2(s)}{R_2} + C_1 s V_1(s) = 0$$

$$\left(\frac{1}{R_2} + C_1 s\right)V_1(s) = \frac{1}{R_2}V_2(s) \quad (2)$$

At Node V_2:

$$\frac{V_2(s) - V_{in}(s)}{R_1} + \frac{V_2(s) - V_1(s)}{R_2} + \frac{V_2(s) - V_{out}(s)}{1/C_2 s} = 0$$

$$\left(\frac{1}{R_1} + \frac{1}{R_2} + C_2 s\right)V_2(s) = \frac{V_{in}(s)}{R_1} + \frac{V_1(s)}{R_2} + C_2 s V_{out}(s) \quad (3)$$

From (1)

$$V_1(s) = \frac{(1/R_4)}{(R_3+R_4)/R_3R_4}V_{out}(s) = \frac{R_3}{(R_3+R_4)}V_{out}(s) \quad (4)$$

From (2)

$$V_2(s) = R_2\left(\frac{1}{R_2}+C_1s\right)V_1(s) = (1+R_2C_1s)V_1(s)$$

and with (4)

$$V_2(s) = \frac{R_3(1+R_2C_1s)}{(R_3+R_4)}V_{out}(s) \quad (5)$$

By substitution of (4) and (5) into (3) we obtain

$$\left(\frac{1}{R_1}+\frac{1}{R_2}+C_2s\right)\frac{R_3(1+R_2C_1s)}{(R_3+R_4)}V_{out}(s) = \frac{V_{in}(s)}{R_1}+\frac{1}{R_2}\frac{R_3}{(R_3+R_4)}V_{out}(s)+C_2sV_{out}(s)$$

$$\left[\left(\frac{1}{R_1}+\frac{1}{R_2}+C_2s\right)\frac{R_3(1+R_2C_1s)}{(R_3+R_4)}-\frac{1}{R_2}\frac{R_3}{(R_3+R_4)}-C_2s\right]V_{out}(s) = \frac{1}{R_1}V_{in}(s)$$

and thus

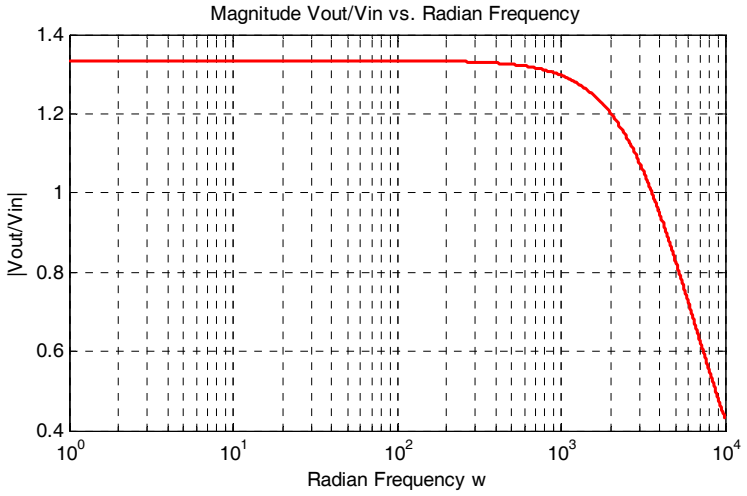
$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{R_1\left[\left(\frac{1}{R_1}+\frac{1}{R_2}+C_2s\right)\frac{R_3(1+R_2C_1s)}{(R_3+R_4)}-\frac{1}{R_2}\frac{R_3}{(R_3+R_4)}-C_2s\right]}$$

By substitution of the given values and after simplification we obtain

$$G(s) = \frac{7.83\times10^7}{s^2+1.77\times10^4s+5.87\times10^7}$$

We use the MATLAB script below to plot this function.

```
w=1:10:10000; s=j.*w; Gs=7.83.*10.^7./(s.^2+1.77.*10.^4.*s+5.87.*10.^7);...
semilogx(w,abs(Gs)); xlabel('Radian Frequency w'); ylabel('|Vout/Vin|');...
title('Magnitude Vout/Vin vs. Radian Frequency'); grid
```



The plot above indicates that this circuit is a second–order low–pass filter.

Chapter 5

State Variables and State Equations

This chapter is an introduction to state variables and state equations as they apply in circuit analysis. The state transition matrix is defined, and the state–space to transfer function equivalence is presented. Several examples are presented to illustrate their application.

5.1 Expressing Differential Equations in State Equation Form

As we know, when we apply Kirchoff's Current Law (KCL) or Kirchoff's Voltage Law (KVL) in networks that contain energy–storing devices, we obtain integro–differential equations. Also, when a network contains just one such device (capacitor or inductor), it is said to be a *first–order circuit*. If it contains two such devices, it is said to be *second–order circuit,* and so on. Thus, a first order linear, time–invariant circuit can be described by a differential equation of the form

$$a_1 \frac{dy}{dt} + a_0\, y(t) = x(t) \tag{5.1}$$

A second order circuit can be described by a second–order differential equation of the same form as (5.1) where the highest order is a second derivative.

An *nth–order* differential equation can be resolved to n first–order simultaneous differential equations with a set of auxiliary variables called *state variables*. The resulting first–order differential equations are called *state–space equations,* or simply *state equations*. These equations can be obtained either from the nth–order differential equation, or directly from the network, provided that the state variables are chosen appropriately. The state variable method offers the advantage that it can also be used with non–linear and time–varying devices. However, our discussion will be limited to linear, time–invariant circuits.

State equations can also be solved with numerical methods such as Taylor series and Runge–Kutta methods, but these will not be discussed in this text*. The state variable method is best illustrated with several examples presented in this chapter.

Example 5.1

A series RLC circuit with excitation

$$v_S(t) = e^{j\omega t} \tag{5.2}$$

* *These are discussed in "Numerical Analysis Using MATLAB and Excel", Third Edition, ISBN 978-1-934404-03-4.*

is described by the integro–differential equation

$$Ri + L\frac{di}{dt} + \frac{1}{C}\int_{-\infty}^{t} i\,dt = e^{j\omega t} \tag{5.3}$$

Differentiating both sides and dividing by L we obtain

$$\frac{d^2t}{dt^2} + \frac{R}{L}\frac{di}{dt} + \frac{1}{LC}i = \frac{1}{L}j\omega e^{j\omega t} \tag{5.4}$$

or

$$\frac{d^2t}{dt^2} = -\frac{R}{L}\frac{di}{dt} - \frac{1}{LC}i + \frac{1}{L}j\omega e^{j\omega t} \tag{5.5}$$

Next, we define two state variables x_1 and x_2 such that

$$x_1 = i \tag{5.6}$$

and

$$x_2 = \frac{di}{dt} = \frac{dx_1}{dt} = \dot{x}_1 \tag{5.7}$$

Then,

$$\dot{x}_2 = d^2i/dt^2 \tag{5.8}$$

where $\dot{x}_k$ denotes the derivative of the state variable x_k. From (5.5) through (5.8), we obtain the state equations

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{R}{L}x_2 - \frac{1}{LC}x_1 + \frac{1}{L}j\omega e^{j\omega t} \end{aligned} \tag{5.9}$$

It is convenient and customary to express the state equations in matrix* form. Thus, we write the state equations of (5.9) as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L}j\omega e^{j\omega t} \end{bmatrix} u \tag{5.10}$$

We usually write (5.10) in a compact form as

$$\dot{x} = Ax + bu \tag{5.11}$$

where

* *For a review of matrix theory, please refer to Appendix D.*

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \frac{1}{L} j\omega e^{j\omega t} \end{bmatrix}, \text{ and } u = \text{any input} \tag{5.12}$$

The output $y(t)$ is expressed by the state equation

$$y = Cx + du \tag{5.13}$$

where C is another matrix, and d is a column vector.

In general, the state representation of a network can be described by the pair of the of the state–space equations

$$\boxed{\begin{aligned} \dot{x} &= Ax + bu \\ y &= Cx + du \end{aligned}} \tag{5.14}$$

The state space equations of (5.14) can be realized with the block diagram of Figure 5.1.

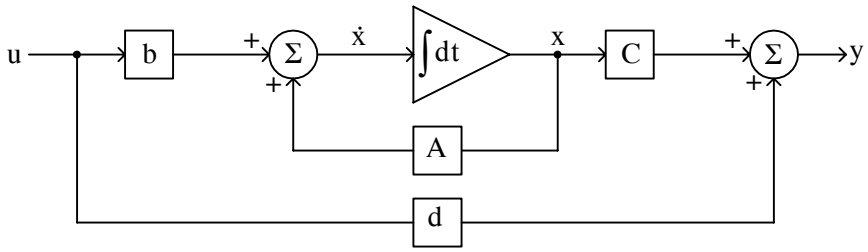


Figure 5.1. Block diagram for the realization of the state equations of (5.14)

We will learn how to solve the matrix equations of (5.14) in the subsequent sections.

Example 5.2

A fourth–order network is described by the differential equation

$$\frac{d^4y}{dt^4} + a_3\frac{d^3y}{dt^3} + a_2\frac{d^2y}{dt^2} + a_1\frac{dy}{dt} + a_0\, y(t) = u(t) \tag{5.15}$$

where $y(t)$ is the output representing the voltage or current of the network, and $u(t)$ is any input. Express (5.15) as a set of state equations.

Solution:

The differential equation of (5.15) is of fourth–order; therefore, we must define four state variables which will be used with the resulting four first–order state equations.

We denote the state variables as x_1, x_2, x_3, and x_4, and we relate them to the terms of the given differential equation as

$$x_1 = y(t) \qquad x_2 = \frac{dy}{dt} \qquad x_3 = \frac{d^2y}{dt^2} \qquad x_4 = \frac{d^3y}{dt^3} \tag{5.16}$$

We observe that

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= x_4 \\ \frac{d^4y}{dt^4} &= \dot{x}_4 = -a_0x_1 - a_1x_2 - a_2x_3 - a_3x_4 + u(t) \end{aligned} \tag{5.17}$$

and in matrix form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t) \tag{5.18}$$

In compact form, (5.18) is written as

$$\dot{x} = Ax + bu \tag{5.19}$$

where

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \text{and } u = u(t)$$

We can also obtain the state equations directly from given circuits. We choose the state variables to represent inductor currents and capacitor voltages. In other words, we assign state variables to energy storing devices. The examples below illustrate the procedure.

Example 5.3

Write state equation(s) for the circuit of Figure 5.2, given that $v_C(0^-) = 0$, and $u_0(t)$ is the unit step function.

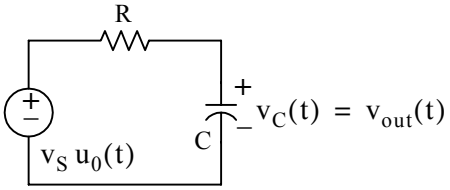


Figure 5.2. Circuit for Example 5.3

Solution:

This circuit contains only one energy–storing device, the capacitor. Therefore, we need only one state variable. We choose the state variable to denote the voltage across the capacitor as shown in Figure 5.3. The output is defined as the voltage across the capacitor.

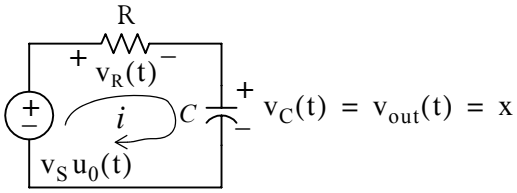


Figure 5.3. Circuit for Example 5.3 with state variable x assigned to it

For this circuit,

$$i_R = i = i_C = C\frac{dv_C}{dt} = C\dot{x}$$

and

$$v_R(t) = Ri = RC\dot{x}$$

By KVL,

$$v_R(t) + v_C(t) = v_S\, u_0(t)$$

or

$$RC\dot{x} + x = v_S u_0(t)$$

Therefore, the state equations are

$$\dot{x} = -\frac{1}{RC}x + v_S\, u_0(t) \qquad (5.20)$$

$$y = x$$

Example 5.4

Write state equation(s) for the circuit of Figure 5.4 assuming $i_L(0^-) = 0$, and the output y is defined as $y = i(t)$.

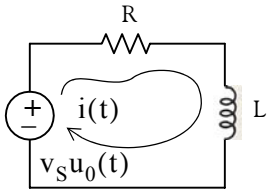


Figure 5.4. Circuit for Example 5.4

Solution:

This circuit contains only one energy–storing device, the inductor; therefore, we need only one state variable. We choose the state variable to denote the current through the inductor as shown in Figure 5.5.

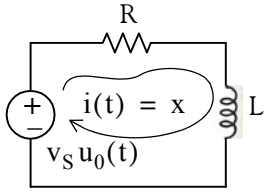


Figure 5.5. Circuit for Example 5.4 with assigned state variable x

By KVL,

$$v_R + v_L = v_S u_0(t)$$

or

$$Ri + L\frac{di}{dt} = v_S u_0(t)$$

or

$$Rx + L\dot{x} = v_S u_0(t)$$

Therefore, the state equations are

$$\begin{aligned} \dot{x} &= -\frac{R}{L}x + \frac{1}{L}v_S u_0(t) \\ y &= x \end{aligned} \tag{5.21}$$

5.2 Solution of Single State Equations

If a circuit contains only one energy–storing device, the state equations are written as

$$\begin{aligned} \dot{x} &= \alpha x + \beta u \\ y &= k_1 x + k_2 u \end{aligned} \tag{5.22}$$

where α, β, k_1, and k_2 are scalar constants, and the initial condition, if non–zero, is denoted as

$$x_0 = x(t_0) \tag{5.23}$$

We will now prove that the solution of the first state equation in (5.22) is

$$x(t) = e^{\alpha(t-t_0)}x_0 + e^{\alpha t}\int_{t_0}^{t} e^{-\alpha\tau}\beta u(\tau)d\tau \tag{5.24}$$

Proof:

First, we must show that (5.24) satisfies the initial condition of (5.23). This is done by substitution of $t = t_0$ in (5.24). Then,

$$x(t_0) = e^{\alpha(t_0-t_0)}x_0 + e^{\alpha t}\int_{t_0}^{t_0} e^{-\alpha\tau}\beta u(\tau)d\tau \tag{5.25}$$

The first term in the right side of (5.25) reduces to x_0 since

$$e^{\alpha(t_0-t_0)}x_0 = e^{0}x_0 = x_0 \tag{5.26}$$

The second term of (5.25) is zero since the upper and lower limits of integration are the same. Therefore, (5.25) reduces to $x(t_0) = x_0$ and thus the initial condition is satisfied.

Next, we must prove that (5.24) satisfies also the first equation in (5.22). To prove this, we differentiate (5.24) with respect to t and we obtain

$$\dot{x}(t) = \frac{d}{dt}(e^{\alpha(t-t_0)}x_0) + \frac{d}{dt}\left\{e^{\alpha t}\int_{t_0}^{t} e^{-\alpha\tau}\beta u(\tau)d\tau\right\}$$

or

$$\dot{x}(t) = \alpha e^{\alpha(t-t_0)}x_0 + \alpha e^{\alpha t}\int_{t_0}^{t} e^{-\alpha\tau}\beta u(\tau)d\tau + e^{\alpha t}[e^{-\alpha\tau}\beta u(\tau)]\Big|_{\tau=t}$$

$$= \alpha\left[e^{\alpha(t-t_0)}x_0 + e^{\alpha t}\int_{t_0}^{t} e^{-\alpha\tau}\beta u(\tau)d\tau\right] + e^{\alpha t}e^{-\alpha t}\beta u(t)$$

or

$$\dot{x}(t) = \alpha\left[e^{\alpha(t-t_0)}x_0 + \int_{t_0}^{t} e^{\alpha(t-\tau)}\beta u(\tau)d\tau\right] + \beta u(t) \tag{5.27}$$

We observe that the bracketed terms of (5.27) are the same as the right side of the assumed solution of (5.24). Therefore,

$$\dot{x} = \alpha x + \beta u$$

and this is the same as the first equation of (5.22).

In summary, if α and β are scalar constants, the solution of

$$\dot{x} = \alpha x + \beta u \tag{5.28}$$

with initial condition

$$x_0 = x(t_0) \tag{5.29}$$

is obtained from the relation

$$x(t) = e^{\alpha(t-t_0)}x_0 + e^{\alpha t}\int_{t_0}^{t} e^{-\alpha\tau}\beta u(\tau)d\tau \tag{5.30}$$

Example 5.5

Use (5.28) through (5.30) to find the capacitor voltage $v_C(t)$ of the circuit of Figure 5.6 for $t > 0$, given that the initial condition is $v_C(0^-) = 1$ V

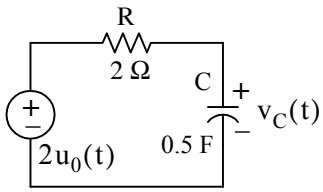


Figure 5.6. Circuit for Example 5.5

Solution:

From (5.20) of Example 5.3, Page 5–5,

$$\dot{x} = -\frac{1}{RC}x + v_S u_0(t)$$

and by comparison with (5.28),

$$\alpha = -\frac{1}{RC} = \frac{-1}{2\times 0.5} = -1$$

and

$$\beta = 2$$

Then, from (5.30),

$$x(t) = e^{\alpha(t-t_0)}x_0 + e^{\alpha t}\int_{t_0}^{t} e^{-\alpha\tau}\beta u(\tau)d\tau = e^{-1(t-0)}1 + e^{-t}\int_{0}^{t} e^{\tau}2u(\tau)d\tau$$

$$= e^{-t} + 2e^{-t}\int_{0}^{t} e^{\tau}d\tau = e^{-t} + 2e^{-t}[e^{\tau}]\Big|_0^t = e^{-t} + 2e^{-t}(e^{t}-1)$$

or

$$v_C(t) = x(t) = (2 - e^{-t})u_0(t) \tag{5.31}$$

Assuming that the output y is the capacitor voltage, the output state equation is

$$y(t) = x(t) = (2 - e^{-t})u_0(t) \tag{5.32}$$

5.3 The State Transition Matrix

In Section 5.1, relation (5.14), we defined the state equations pair

$$\begin{aligned} \dot{x} &= Ax + bu \\ y &= Cx + du \end{aligned} \tag{5.33}$$

where for two or more simultaneous differential equations, A and C are 2×2 or higher order matrices, and b and d are column vectors with two or more rows. In this section we will introduce the *state transition matrix* e^{At}, and we will prove that the solution of the matrix differential equation

$$\dot{x} = Ax + bu \tag{5.34}$$

with initial conditions

$$x(t_0) = x_0 \tag{5.35}$$

is obtained from the relation

$$x(t) = e^{A(t-t_0)}x_0 + e^{At}\int_{t_0}^{t} e^{-A\tau}bu(\tau)d\tau \tag{5.36}$$

Proof:

Let A be any $n \times n$ matrix whose elements are constants. Then, another $n \times n$ matrix denoted as $\varphi(t)$, is said to be the state transition matrix of (5.34), if it is related to the matrix A as the matrix power series

$$\boxed{\varphi(t) \equiv e^{At} = I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \ldots + \frac{1}{n!}A^nt^n} \tag{5.37}$$

where I is the $n \times n$ identity matrix.

From (5.37), we find that

$$\varphi(0) = e^{A0} = I + A0 + \ldots = I \tag{5.38}$$

Differentiation of (5.37) with respect to t yields

$$\varphi'(t) = \frac{d}{dt}e^{At} = 0 + A \cdot 1 + A^2t + \ldots = A + A^2t + \ldots \tag{5.39}$$

and by comparison with (5.37) we obtain

$$\frac{d}{dt}e^{At} = Ae^{At} \tag{5.40}$$

To prove that (5.36) is the solution of (5.34), we must prove that it satisfies both the initial condition and the matrix differential equation. The initial condition is satisfied from the relation

$$x(t_0) = e^{A(t_0-t_0)}x_0 + e^{At_0}\int_{t_0}^{t_0} e^{-A\tau}bu(\tau)d\tau = e^{A0}x_0 + 0 = Ix_0 = x_0 \tag{5.41}$$

where we have used (5.38) for the initial condition. The integral is zero since the upper and lower limits of integration are the same.

To prove that (5.34) is also satisfied, we differentiate the assumed solution

$$x(t) = e^{A(t-t_0)}x_0 + e^{At}\int_{t_0}^{t} e^{-A\tau}bu(\tau)d\tau$$

with respect to t and we use (5.40), that is,

$$\frac{d}{dt}e^{At} = Ae^{At}$$

Then,

$$\dot{x}(t) = Ae^{A(t-t_0)}x_0 + Ae^{At}\int_{t_0}^{t} e^{-A\tau}bu(\tau)d\tau + e^{At}e^{-At}bu(t)$$

or

$$\dot{x}(t) = A\left[e^{A(t-t_0)}x_0 + e^{At}\int_{t_0}^{t} e^{-A\tau}bu(\tau)d\tau\right] + e^{At}e^{-At}bu(t) \tag{5.42}$$

We recognize the bracketed terms in (5.42) as $x(t)$, and the last term as $bu(t)$. Thus, the expression (5.42) reduces to

$$\dot{x}(t) = Ax + bu$$

In summary, if A is an $n \times n$ matrix whose elements are constants, $n \geq 2$, and b is a column vector with *n* elements, the solution of

$$\dot{x}(t) = Ax + bu \tag{5.43}$$

with initial condition

$$x_0 = x(t_0) \tag{5.44}$$

is

$$\boxed{x(t) = e^{A(t-t_0)}x_0 + e^{At}\int_{t_0}^{t} e^{-A\tau}bu(\tau)d\tau} \tag{5.45}$$

Therefore, the solution of second or higher order circuits using the state variable method, entails the computation of the state transition matrix e^{At}, and integration of (5.45).

5.4 Computation of the State Transition Matrix e^{At}

Let A be an $n \times n$ matrix, and I be the $n \times n$ identity matrix. By definition, the *eigenvalues* λ_i, $i = 1, 2, \ldots, n$ of A are the roots of the nth order polynomial

$$\det[A - \lambda I] = 0 \tag{5.46}$$

We recall that expansion of a determinant produces a polynomial. The roots of the polynomial of (5.46) can be real (unequal or equal), or complex numbers.

Evaluation of the state transition matrix e^{At} is based on the *Cayley–Hamilton theorem*. This theorem states that a matrix can be expressed as an $(n-1)$th degree polynomial in terms of the matrix A as

$$e^{At} = a_0 I + a_1 A + a_2 A^2 + \ldots + a_{n-1} A^{n-1} \tag{5.47}$$

where the coefficients a_i are functions of the eigenvalues λ.

We accept (5.47) without proving it. The proof can be found in Linear Algebra and Matrix Theory textbooks.

Since the coefficients a_i are functions of the eigenvalues λ, we must consider the two cases discussed in Subsections 5.4.1 and 5.4.2 below.

5.4.1 Distinct Eigenvalues (Real of Complex)

If $\lambda_1 \neq \lambda_2 \neq \lambda_3 \neq \ldots \neq \lambda_n$, that is, if all eigenvalues of a given matrix A are distinct, the coefficients a_i are found from the simultaneous solution of the following system of equations:

$$\begin{aligned}
a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1} &= e^{\lambda_1 t} \\
a_0 + a_1\lambda_2 + a_2\lambda_2^2 + \ldots + a_{n-1}\lambda_2^{n-1} &= e^{\lambda_2 t} \\
&\ldots \\
a_0 + a_1\lambda_n + a_2\lambda_n^2 + \ldots + a_{n-1}\lambda_n^{n-1} &= e^{\lambda_n t}
\end{aligned} \tag{5.48}$$

Example 5.6

Compute the state transition matrix e^{At} given that

$$A = \begin{bmatrix} -2 & 1 \\ 0 & -1 \end{bmatrix}$$

Solution:

We must first find the eigenvalues λ of the given matrix A. These are found from the expansion of

$$\det[A - \lambda I] = 0$$

For this example,

$$\det[A - \lambda I] = \det\left\{\begin{bmatrix} -2 & 1 \\ 0 & -1 \end{bmatrix} - \lambda\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right\} = \det\begin{bmatrix} -2-\lambda & 1 \\ 0 & -1-\lambda \end{bmatrix} = 0$$

$$= (-2-\lambda)(-1-\lambda) = 0$$

or

$$(\lambda + 1)(\lambda + 2) = 0$$

Therefore,

$$\lambda_1 = -1 \quad \text{and} \quad \lambda_2 = -2 \tag{5.49}$$

Next, we must find the coefficients a_i of (5.47). Since A is a 2×2 matrix, we only need to consider the first two terms of that relation, that is,

$$e^{At} = a_0 I + a_1 A \tag{5.50}$$

The coefficients a_0 and a_1 are found from (5.48). For this example,

$$a_0 + a_1\lambda_1 = e^{\lambda_1 t}$$

$$a_0 + a_1\lambda_2 = e^{\lambda_2 t}$$

or

$$\begin{aligned} a_0 + a_1(-1) &= e^{-t} \\ a_0 + a_1(-2) &= e^{-2t} \end{aligned} \tag{5.51}$$

Simultaneous solution of (5.51) yields

$$\begin{aligned} a_0 &= 2e^{-t} - e^{-2t} \\ a_1 &= e^{-t} - e^{-2t} \end{aligned} \tag{5.52}$$

and by substitution into (5.50),

$$e^{At} = (2e^{-t} - e^{-2t})\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + (e^{-t} - e^{-2t})\begin{bmatrix} -2 & 1 \\ 0 & -1 \end{bmatrix}$$

or

$$e^{At} = \begin{bmatrix} e^{-2t} & e^{-t} - e^{-2t} \\ 0 & e^{-t} \end{bmatrix} \tag{5.53}$$

In summary, we compute the state transition matrix e^{At} for a given matrix A using the following procedure:

1. We find the eigenvalues λ from $\det[A - \lambda I] = 0$. We can write $[A - \lambda I]$ at once by subtracting λ from each of the main diagonal elements of A. If the dimension of A is a 2×2 matrix, it will yield two eigenvalues; if it is a 3×3 matrix, it will yield three eigenvalues, and so on. If the eigenvalues are distinct, we perform steps 2 through 4; otherwise we refer to Subsection 5.4.2 below.

2. If the dimension of A is a 2×2 matrix, we use only the first *2* terms of the right side of the state transition matrix

$$e^{At} = a_0 I + a_1 A + a_2 A^2 + \ldots + a_{n-1} A^{n-1} \tag{5.54}$$

   If A matrix is a 3×3 matrix, we use the first 3 terms of (5.54), and so on.

3. We obtain the a_i coefficients from

$$\begin{aligned} a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1} &= e^{\lambda_1 t} \\ a_0 + a_1\lambda_2 + a_2\lambda_2^2 + \ldots + a_{n-1}\lambda_2^{n-1} &= e^{\lambda_2 t} \\ \ldots & \\ a_0 + a_1\lambda_n + a_2\lambda_n^2 + \ldots + a_{n-1}\lambda_n^{n-1} &= e^{\lambda_n t} \end{aligned}$$

   We use as many equations as the number of the eigenvalues, and we solve for the coefficients a_i.

4. We substitute the a_i coefficients into the state transition matrix of (5.54), and we simplify.

Example 5.7

Compute the state transition matrix e^{At} given that

$$A = \begin{bmatrix} 5 & 7 & -5 \\ 0 & 4 & -1 \\ 2 & 8 & -3 \end{bmatrix} \tag{5.55}$$

Solution:

1. We first compute the eigenvalues from $\det[A - \lambda I] = 0$. We obtain $[A - \lambda I]$ at once, by subtracting λ from each of the main diagonal elements of A. Then,

$$\det[A - \lambda I] = \det\begin{bmatrix} 5-\lambda & 7 & -5 \\ 0 & 4-\lambda & -1 \\ 2 & 8 & -3-\lambda \end{bmatrix} = 0 \tag{5.56}$$

and expansion of this determinant yields the polynomial

$$\lambda^3 - 6\lambda^2 + 11\lambda - 6 = 0 \tag{5.57}$$

We will use MATLAB **roots(p)** function to obtain the roots of (5.57).

```
p=[1 −6 11 −6]; r=roots(p); fprintf(' \n'); fprintf('lambda1 = %5.2f \t', r(1));...
fprintf('lambda2 = %5.2f \t', r(2)); fprintf('lambda3 = %5.2f', r(3))
```

```
lambda1 = 3.00    lambda2 = 2.00    lambda3 = 1.00
```

and thus the eigenvalues are

$$\lambda_1 = 1 \qquad \lambda_2 = 2 \qquad \lambda_3 = 3 \tag{5.58}$$

2. Since A is a 3×3 matrix, we use the first 3 terms of (5.54), that is,

$$e^{At} = a_0 I + a_1 A + a_2 A^2 \tag{5.59}$$

3. We obtain the coefficients a_0, a_1, and a_2 from

$$\begin{aligned} a_0 + a_1\lambda_1 + a_2\lambda_1^2 &= e^{\lambda_1 t} \\ a_0 + a_1\lambda_2 + a_2\lambda_2^2 &= e^{\lambda_2 t} \\ a_0 + a_1\lambda_3 + a_2\lambda_3^2 &= e^{\lambda_3 t} \end{aligned}$$

or

$$\begin{aligned} a_0 + a_1 + a_2 &= e^{t} \\ a_0 + 2a_1 + 4a_2 &= e^{2t} \\ a_0 + 3a_1 + 9a_2 &= e^{3t} \end{aligned} \tag{5.60}$$

We will use the following MATLAB script for the solution of (5.60).

```
B=sym('[1  1  1; 1  2  4; 1  3  9]'); b=sym('[exp(t); exp(2*t); exp(3*t)]'); a=B\b; fprintf(' \n');...
disp('a0 = '); disp(a(1)); disp('a1 = '); disp(a(2)); disp('a2 = '); disp(a(3))
```

```
a0 =
3*exp(t)-3*exp(2*t)+exp(3*t)
a1 =
-5/2*exp(t)+4*exp(2*t)-3/2*exp(3*t)
a2 =
1/2*exp(t)-exp(2*t)+1/2*exp(3*t)
```

Thus,

$$\begin{aligned} a_0 &= 3e^{t} - 3e^{2t} + e^{3t} \\ a_1 &= -\frac{5}{2}e^{t} + 4e^{2t} - \frac{3}{2}e^{3t} \\ a_2 &= \frac{1}{2}e^{t} - e^{2t} + \frac{1}{2}e^{3t} \end{aligned} \tag{5.61}$$

4. We also use MATLAB to perform the substitution into the state transition matrix, and to perform the matrix multiplications. The script is shown below.

```
syms t; a0 = 3*exp(t)+exp(3*t)–3*exp(2*t); a1 = –5/2*exp(t)–3/2*exp(3*t)+4*exp(2*t);...
a2 = 1/2*exp(t)+1/2*exp(3*t)–exp(2*t);...
A = [5  7  –5;  0  4  –1;  2  8  -3]; eAt=a0*eye(3)+a1*A+a2*A^2
```

```
eAt =
[-2*exp(t)+2*exp(2*t)+exp(3*t),   -6*exp(t)+5*exp(2*t)+exp(3*t),
4*exp(t)-3*exp(2*t)-exp(3*t)]
[-exp(t)+2*exp(2*t)-exp(3*t),   -3*exp(t)+5*exp(2*t)-exp(3*t),
2*exp(t)-3*exp(2*t)+exp(3*t)]
[-3*exp(t)+4*exp(2*t)-exp(3*t), -9*exp(t)+10*exp(2*t)-exp(3*t),
6*exp(t)-6*exp(2*t)+exp(3*t)]
```

Thus,

$$e^{At} = \begin{bmatrix} -2e^{t} + 2e^{2t} + e^{3t} & -6e^{t} + 5e^{2t} + e^{3t} & 4e^{t} - 3e^{2t} - e^{3t} \\ -e^{t} + 2e^{2t} - e^{3t} & -3e^{t} + 5e^{2t} - e^{3t} & 2e^{t} - 3e^{2t} + e^{3t} \\ -3e^{t} + 4e^{2t} - e^{3t} & -9e^{t} + 10e^{2t} - e^{3t} & 6e^{t} - 6e^{2t} + e^{3t} \end{bmatrix}$$

5.4.2 Multiple (Repeated) Eigenvalues

In this case, we will assume that the polynomial of

$$\det[A - \lambda I] = 0 \tag{5.62}$$

has n roots, and m of these roots are equal. In other words, the roots are

$$\lambda_1 = \lambda_2 = \lambda_3 \ldots = \lambda_m,\ \lambda_{m+1},\ \lambda_n \tag{5.63}$$

The coefficients a_i of the state transition matrix

$$e^{At} = a_0 I + a_1 A + a_2 A^2 + \ldots + a_{n-1} A^{n-1} \tag{5.64}$$

are found from the simultaneous solution of the system of equations of (5.65) below.

$$\begin{aligned}
a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1} &= e^{\lambda_1 t} \\
\frac{d}{d\lambda_1}(a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1}) &= \frac{d}{d\lambda_1}e^{\lambda_1 t} \\
\frac{d^2}{d\lambda_1^2}(a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1}) &= \frac{d^2}{d\lambda_1^2}e^{\lambda_1 t} \\
&\ldots \\
\frac{d^{m-1}}{d\lambda_1^{m-1}}(a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1}) &= \frac{d^{m-1}}{d\lambda_1^{m-1}}e^{\lambda_1 t} \\
a_0 + a_1\lambda_{m+1} + a_2\lambda_{m+1}^2 + \ldots + a_{n-1}\lambda_{m+1}^{n-1} &= e^{\lambda_{m+1} t} \\
&\ldots \\
a_0 + a_1\lambda_n + a_2\lambda_n^2 + \ldots + a_{n-1}\lambda_n^{n-1} &= e^{\lambda_n t}
\end{aligned} \tag{5.65}$$

Example 5.8

Compute the state transition matrix e^{At} given that

$$A = \begin{bmatrix} -1 & 0 \\ 2 & -1 \end{bmatrix}$$

Solution:

1. We first find the eigenvalues λ of the matrix A and these are found from the polynomial of $\det[A - \lambda I] = 0$. For this example,

$$\det[A - \lambda I] = \det\begin{bmatrix} -1-\lambda & 0 \\ 2 & -1-\lambda \end{bmatrix} = 0 \qquad (-1-\lambda)(-1-\lambda) = 0 \qquad (\lambda+1)^2 = 0$$

and thus,

$$\lambda_1 = \lambda_2 = -1$$

2. Since A is a 2×2 matrix, we only need the first two terms of the state transition matrix, that is,

$$e^{At} = a_0 I + a_1 A \tag{5.66}$$

3. We find a_0 and a_1 from (5.65). For this example,

$$a_0 + a_1\lambda_1 = e^{\lambda_1 t}$$

$$\frac{d}{d\lambda_1}(a_0 + a_1\lambda_1) = \frac{d}{d\lambda_1}e^{\lambda_1 t}$$

or

$$a_0 + a_1\lambda_1 = e^{\lambda_1 t}$$

$$a_1 = te^{\lambda_1 t}$$

and by substitution with $\lambda_1 = \lambda_2 = -1$, we obtain

$$a_0 - a_1 = e^{-t}$$

$$a_1 = te^{-t}$$

Simultaneous solution of the last two equations yields

$$\begin{aligned} a_0 &= e^{-t} + te^{-t} \\ a_1 &= te^{-t} \end{aligned} \tag{5.67}$$

4. By substitution of (5.67) into (5.66), we obtain

$$e^{At} = (e^{-t} + te^{-t})\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + te^{-t}\begin{bmatrix} -1 & 0 \\ 2 & -1 \end{bmatrix}$$

or

$$e^{At} = \begin{bmatrix} e^{-t} & 0 \\ 2te^{-t} & e^{-t} \end{bmatrix} \tag{5.68}$$

We can use the MATLAB **eig(x)** function to find the eigenvalues of an $n \times n$ matrix. To find out how it is used, we invoke the **help eig** command.

We will first use MATLAB to verify the values of the eigenvalues found in Examples 5.6 through 5.8, and we will briefly discuss eigenvectors in the next section.

Example 5.6:

```
A= [–2  1; 0  –1]; lambda=eig(A)
```

```
lambda =
    -2
    -1
```

Example 5.7:

```
B = [5  7  –5;  0  4  –1;  2  8  –3]; lambda=eig(B)
```

```
lambda =
    1.0000
    3.0000
    2.0000
```

Example 5.8:

```
C = [–1  0; 2  –1]; lambda=eig(C)
```

```
lambda =
    -1
    -1
```

5.5 Eigenvectors

Consider the relation

$$AX = \lambda X \tag{5.69}$$

where A is an $n \times n$ matrix, X is a column vector, and λ is a scalar number. We can express this relation in matrix form as

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \tag{5.70}$$

We write (5.70) as

$$(A-\lambda I)X = 0 \tag{5.71}$$

Then, (5.71) can be written as

$$\begin{bmatrix} (a_{11}-\lambda)x_1 & a_{12}x_2 & \dots & a_{1n}x_n \\ a_{21}x_1 & (a_{22}-\lambda)x_2 & \dots & a_{2n}x_n \\ \dots & \dots & \dots & \dots \\ a_{n1}x_1 & a_{n2}x_2 & \dots & (a_{nn}-\lambda)x_n \end{bmatrix} = 0 \tag{5.72}$$

The equations of (5.72) will have non–trivial solutions if and only if its determinant is zero*, that is, if

$$\det\begin{bmatrix} (a_{11}-\lambda) & a_{12} & \dots & a_{1n} \\ a_{21} & (a_{22}-\lambda) & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & (a_{nn}-\lambda) \end{bmatrix} = 0 \tag{5.73}$$

Expansion of the determinant of (5.73) results in a polynomial equation of degree n in λ, and it is called the *characteristic equation*.

We can express (5.73) in a compact form as

$$\det(A-\lambda I) = 0 \tag{5.74}$$

As we know, the roots λ of the characteristic equation are the eigenvalues of the matrix A, and corresponding to each eigenvalue λ, there is a non-trivial solution of the column vector X, i.e., $X \neq 0$. This vector X is called *eigenvector*. Obviously, there is a different eigenvector for each eigenvalue. Eigenvectors are generally expressed as *unit eigenvectors*, that is, they are normalized to unit length. This is done by dividing each component of the eigenvector by the square root of the sum of the squares of their components, so that the sum of the squares of their components is equal to unity.

In many engineering applications the unit eigenvectors are chosen such that $X \cdot X^T = I$ where X^T is the transpose of the eigenvector X, and I is the identity matrix.

Two vectors X and Y are said to be *orthogonal* if their inner (dot) product is zero. A set of eigenvectors constitutes an *orthonormal basis* if the set is normalized (expressed as unit eigenvectors) and these vector are mutually orthogonal. An orthonormal basis can be formed with the *Gram-Schmidt Orthogonalization Procedure*; it is beyond the scope of this chapter to discuss this procedure, and therefore it will not be discussed in this text. It can be found in Linear Algebra and Matrix Theory textbooks.

* *This is because we want the vector X in (5.71) to be a non-zero vector and the product* $(A-\lambda I)X$ *to be zero.*

The example below illustrates the relationships between a matrix A, its eigenvalues, and eigenvectors.

Example 5.9

Given the matrix

$$A = \begin{bmatrix} 5 & 7 & -5 \\ 0 & 4 & -1 \\ 2 & 8 & -3 \end{bmatrix}$$

a. Find the eigenvalues of A

b. Find eigenvectors corresponding to each eigenvalue of A

c. Form a set of unit eigenvectors using the eigenvectors of part (b).

Solution:

a. This is the same matrix as in Example 5.7, relation (5.55), Page 5–14, where we found the eigenvalues to be

$$\lambda_1 = 1 \qquad \lambda_2 = 2 \qquad \lambda_3 = 3$$

b. We start with

$$AX = \lambda X$$

and we let

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Then,

$$\begin{bmatrix} 5 & 7 & -5 \\ 0 & 4 & -1 \\ 2 & 8 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{5.75}$$

or

$$\begin{bmatrix} 5x_1 & 7x_2 & -5x_3 \\ 0 & 4x_2 & -x_3 \\ 2x_1 & 8x_2 & -3x_3 \end{bmatrix} = \begin{bmatrix} \lambda x_1 \\ \lambda x_2 \\ \lambda x_3 \end{bmatrix} \tag{5.76}$$

Equating corresponding rows and rearranging, we obtain

$$\begin{bmatrix} (5-\lambda)x_1 & 7x_2 & -5x_3 \\ 0 & (4-\lambda)x_2 & -x_3 \\ 2x_1 & 8x_2 & -(3-\lambda)x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.77}$$

For $\lambda = 1$, (5.77) reduces to

$$\begin{aligned} 4x_1 + 7x_2 - 5x_3 &= 0 \\ 3x_2 - x_3 &= 0 \\ 2x_1 + 8x_2 - 4x_3 &= 0 \end{aligned} \tag{5.78}$$

By Crame's rule, or MATLAB, we obtain the indeterminate values

$$x_1 = 0/0 \qquad x_2 = 0/0 \qquad x_3 = 0/0 \tag{5.79}$$

Since the unknowns $x_1, x_2,$ and x_3 are scalars, we can assume that one of these, say x_2, is known, and solve x_1 and x_3 in terms of x_2. Then, we obtain $x_1 = 2x_2$, and $x_3 = 3x_2$. Therefore, an eigenvector for $\lambda = 1$ is

$$X_{\lambda=1} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2x_2 \\ x_2 \\ 3x_2 \end{bmatrix} = x_2 \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} \tag{5.80}$$

since any eigenvector is a scalar multiple of the last vector in (5.80).

Similarly, for $\lambda = 2$, we obtain $x_1 = x_2$, and $x_3 = 2x_2$. Then, an eigenvector for $\lambda = 2$ is

$$X_{\lambda=2} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_2 \\ 2x_2 \end{bmatrix} = x_2 \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \tag{5.81}$$

Finally, for $\lambda = 3$, we obtain $x_1 = -x_2$, and $x_3 = x_2$. Then, an eigenvector for $\lambda = 3$ is

$$X_{\lambda=3} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -x_2 \\ x_2 \\ x_2 \end{bmatrix} = x_2 \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} \tag{5.82}$$

c. We find the unit eigenvectors by dividing the components of each vector by the square root of the sum of the squares of the components. These are:

$$\sqrt{2^2+1^2+3^2} = \sqrt{14}$$

$$\sqrt{1^2+1^2+2^2} = \sqrt{6}$$

$$\sqrt{(-1)^2+1^2+1^2} = \sqrt{3}$$

The unit eigenvectors are

$$\text{Unit } X_{\lambda=1} = \begin{bmatrix} \frac{2}{\sqrt{14}} \\ \frac{1}{\sqrt{14}} \\ \frac{3}{\sqrt{14}} \end{bmatrix} \qquad \text{Unit } X_{\lambda=2} = \begin{bmatrix} \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \end{bmatrix} \qquad \text{Unit } X_{\lambda=3} = \begin{bmatrix} \frac{-1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix} \tag{5.83}$$

We observe that for the first unit eigenvector the sum of the squares is unity, that is,

$$\left(\frac{2}{\sqrt{14}}\right)^2 + \left(\frac{1}{\sqrt{14}}\right)^2 + \left(\frac{3}{\sqrt{14}}\right)^2 = \frac{4}{14} + \frac{1}{14} + \frac{9}{14} = 1 \tag{5.84}$$

and the same is true for the other two unit eigenvectors in (5.83).

5.6 Circuit Analysis with State Variables

In this section we will present two examples to illustrate how the state variable method is used in circuit analysis.

Example 5.10

For the circuit of Figure 5.7, the initial conditions are $i_L(0^-) = 0$, and $v_C(0^-) = 0.5$ V. Use the state variable method to compute $i_L(t)$ and $v_C(t)$.

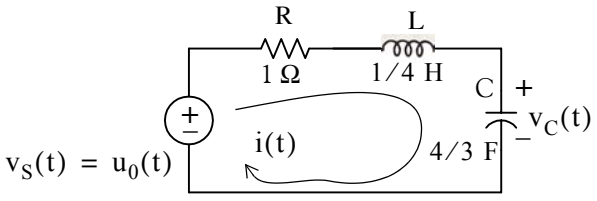


Figure 5.7. Circuit for Example 5.10

Solution:

For this example,

$$i = i_L$$

and

$$Ri_L + L\frac{di_L}{dt} + v_C = u_0(t)$$

Substitution of given values and rearranging, yields

$$\frac{1}{4}\frac{di_L}{dt} = (-1)i_L - v_C + 1$$

or

$$\frac{di_L}{dt} = -4i_L - 4v_C + 4 \tag{5.85}$$

Next, we define the state variables $x_1 = i_L$ and $x_2 = v_C$. Then,

$$\dot{x}_1 = \frac{di_L}{dt} \tag{5.86}$$

and

$$\dot{x}_2 = \frac{dv_C}{dt}$$

Also,

$$i_L = C\frac{dv_C}{dt}$$

and thus,

$$x_1 = i_L = C\frac{dv_C}{dt} = C\dot{x}_2 = \frac{4}{3}\dot{x}_2$$

or

$$\dot{x}_2 = \frac{3}{4}x_1 \tag{5.87}$$

Therefore, from (5.85), (5.86), and (5.87), we obtain the state equations

$$\dot{x}_1 = -4x_1 - 4x_2 + 4$$

$$\dot{x}_2 = \frac{3}{4}x_1$$

and in matrix form,

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-4 & -4\\ 3/4 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}4\\ 0\end{bmatrix}u_0(t) \tag{5.88}$$

We will compute the solution of (5.88) using

$$x(t) = e^{A(t-t_0)}x_0 + e^{At}\int_{t_0}^{t} e^{-A\tau}bu(\tau)d\tau \tag{5.89}$$

where

$$A = \begin{bmatrix} -4 & -4 \\ 3/4 & 0 \end{bmatrix} \quad x_0 = \begin{bmatrix} i_L(0) \\ v_C(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \end{bmatrix} \quad b = \begin{bmatrix} 4 \\ 0 \end{bmatrix} \tag{5.90}$$

First, we compute the state transition matrix e^{At}. We find the eigenvalues from

$$\det[A - \lambda I] = 0$$

Then,

$$\det[A - \lambda I] = \det\begin{bmatrix} -4-\lambda & -4 \\ 3/4 & -\lambda \end{bmatrix} = 0 \qquad (-\lambda)(-4-\lambda) + 3 = 0 \qquad \lambda^2 + 4\lambda + 3 = 0$$

Therefore,

$$\lambda_1 = -1 \text{ and } \lambda_2 = -3$$

The next step is to find the coefficients a_i. Since A is a 2×2 matrix, we only need the first two terms of the state transition matrix, that is,

$$e^{At} = a_0 I + a_1 A \tag{5.91}$$

The constants a_0 and a_1 are found from

$$\begin{aligned} a_0 + a_1\lambda_1 &= e^{\lambda_1 t} \\ a_0 + a_1\lambda_2 &= e^{\lambda_2 t} \end{aligned}$$

and with $\lambda_1 = -1$ and $\lambda_2 = -3$, we obtain

$$\begin{aligned} a_0 - a_1 &= e^{-t} \\ a_0 - 3a_1 &= e^{-3t} \end{aligned} \tag{5.92}$$

Simultaneous solution of (5.92) yields

$$\begin{aligned} a_0 &= 1.5e^{-t} - 0.5e^{-3t} \\ a_1 &= 0.5e^{-t} - 0.5e^{-3t} \end{aligned} \tag{5.93}$$

We now substitute these values into (5.91), and we obtain

$$e^{At} = (1.5e^{-t} - 0.5e^{-3t})\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + (0.5e^{-t} - 0.5e^{-2t})\begin{bmatrix}-4 & -4\\ 3/4 & 0\end{bmatrix}$$

$$= \begin{bmatrix}1.5e^{-t} - 0.5e^{-3t} & 0\\ 0 & 1.5e^{-t} - 0.5e^{-3t}\end{bmatrix} + \begin{bmatrix}-2e^{-t} + 2e^{-3t} & -2e^{-t} + 2e^{-3t}\\ \frac{3}{8}e^{-t} - \frac{3}{8}e^{-3t} & 0\end{bmatrix}$$

or

$$e^{At} = \begin{bmatrix}-0.5e^{-t} + 1.5e^{-3t} & -2e^{-t} + 2e^{-3t}\\ \frac{3}{8}e^{-t} - \frac{3}{8}e^{-3t} & 1.5e^{-t} - 0.5e^{-3t}\end{bmatrix}$$

The initial conditions vector is the second vector in (5.90); then, the first term of (5.89) becomes

$$e^{At}x_0 = \begin{bmatrix}-0.5e^{-t} + 1.5e^{-3t} & -2e^{-t} + 2e^{-3t}\\ \frac{3}{8}e^{-t} - \frac{3}{8}e^{-3t} & 1.5e^{-t} - 0.5e^{-3t}\end{bmatrix}\begin{bmatrix}0\\ 1/2\end{bmatrix}$$

or

$$e^{At}x_0 = \begin{bmatrix}-e^{-t} + e^{-3t}\\ 0.75e^{-t} - 0.25e^{-3t}\end{bmatrix} \tag{5.94}$$

We also need to evaluate the integral on the right side of (5.89). From (5.90)

$$b = \begin{bmatrix}4\\ 0\end{bmatrix} = \begin{bmatrix}1\\ 0\end{bmatrix}4$$

and denoting this integral as Int, we obtain

$$\text{Int} = \int_{t_0}^{t}\begin{bmatrix}-0.5e^{-(t-\tau)} + 1.5e^{-3(t-\tau)} & -2e^{-(t-\tau)} + 2e^{-3(t-\tau)}\\ \frac{3}{8}e^{-(t-\tau)} - \frac{3}{8}e^{-3(t-\tau)} & 1.5e^{-(t-\tau)} - 0.5e^{-3(t-\tau)}\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix}4d\tau$$

or

$$\text{Int} = \int_{t_0}^{t}\begin{bmatrix}-0.5e^{-(t-\tau)} + 1.5e^{-3(t-\tau)}\\ \frac{3}{8}e^{-(t-\tau)} - \frac{3}{8}e^{-3(t-\tau)}\end{bmatrix}4d\tau \tag{5.95}$$

The integration in (5.95) is with respect to τ; then, integrating the column vector under the integral, we obtain

$$\text{Int} = 4\begin{bmatrix} -0.5e^{-(t-\tau)} + 0.5e^{-3(t-\tau)} \\ 0.375e^{-(t-\tau)} - 0.125e^{-3(t-\tau)} \end{bmatrix}\Bigg|_{\tau=0}^{t}$$

or

$$\text{Int} = 4\begin{bmatrix} -0.5+0.5 \\ 0.375-0.125 \end{bmatrix} - 4\begin{bmatrix} -0.5e^{-t} + 0.5e^{-3t} \\ 0.375e^{-t} - 0.125e^{-3t} \end{bmatrix} = 4\begin{bmatrix} 0.5e^{-t} - 0.5e^{-3t} \\ 0.25 - 0.375e^{-t} + 0.125e^{-3t} \end{bmatrix}$$

By substitution of these values, the solution of

$$x(t) = e^{A(t-t_0)}x_0 + e^{At}\int_{t_0}^{t} e^{-A\tau}bu(\tau)d\tau$$

is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -e^{-t} + e^{-3t} \\ 0.75e^{-t} - 0.25e^{-3t} \end{bmatrix} + 4\begin{bmatrix} 0.5e^{-t} - 0.5e^{-3t} \\ 0.25 - 0.375e^{-t} + 0.125e^{-3t} \end{bmatrix} = \begin{bmatrix} e^{-t} - e^{-3t} \\ 1 - 0.75e^{-t} + 0.25e^{-3t} \end{bmatrix}$$

Then,

$$x_1 = i_L = e^{-t} - e^{-3t} \tag{5.96}$$

and

$$x_2 = v_C = 1 - 0.75e^{-t} + 0.25e^{-3t} \tag{5.97}$$

Other variables of the circuit can now be computed from (5.96) and (5.97). For example, the voltage across the inductor is

$$v_L = L\frac{di_L}{dt} = \frac{1}{4}\frac{d}{dt}(e^{-t} - e^{-3t}) = -\frac{1}{4}e^{-t} + \frac{3}{4}e^{-3t}$$

We use the MATLAB script below to plot the relation of (5.97).

```
t=0:0.01:10; x2=1-0.75.*exp(-t)+0.25.*exp(-3.*t);...
plot(t,x2); grid
```

The plot is shown in Figure 5.8.

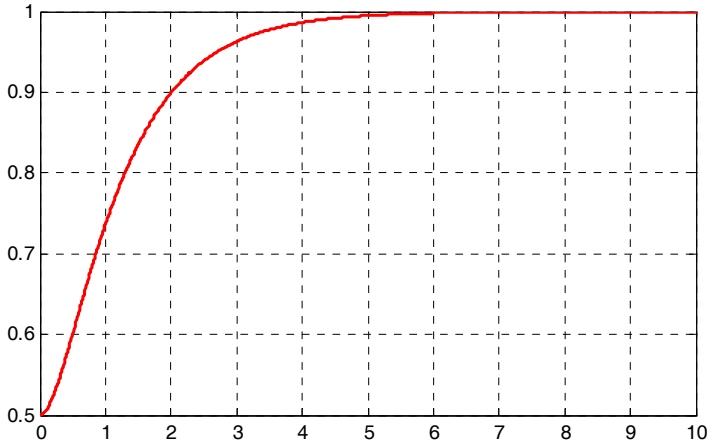


Figure 5.8. Plot for relation (5.97)

We can obtain the plot of Figure 5.8 with the Simulink State–Space block with the unit step function as the input using the **Step** block, and the capacitor voltage as the output displayed on the **Scope** block as shown in the model of Figure 5.9 where for the **State–Space** block Function Block Parameters dialog box we have entered:

A: [–4 –4; 3/4 0]
B: [4 0]'
C: [0 1]
D: [0]
Initial conditions: [0 1/2]

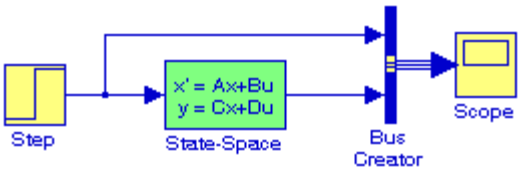


Figure 5.9. Simulink model for Example 5.10

The waveform for the capacitor voltage for the simulation time interval $0 \le t \le 10$ seconds is shown in Figure 5.10 where we observe that the initial condition $v_C(0^-) = 0.5$ V is also displayed.

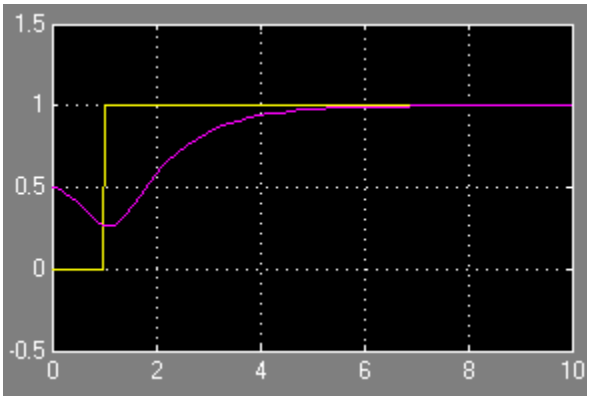


Figure 5.10. Input and output waveforms for the model of Figure 5.9

Example 5.11

A network is described by the state equation

$$\dot{x} = Ax + bu \tag{5.98}$$

where

$$A = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \qquad x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad b = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \text{and} \quad u = \delta(t) \tag{5.99}$$

Compute the state vector

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Solution:

We compute the eigenvalues from

$$\det[A - \lambda I] = 0$$

For this example,

$$\det[A - \lambda I] = \det\begin{bmatrix} 1-\lambda & 0 \\ 1 & -1-\lambda \end{bmatrix} = 0 \qquad (1-\lambda)(-1-\lambda) = 0$$

Then,

$$\lambda_1 = 1 \quad \text{and} \quad \lambda_2 = -1$$

Since A is a 2×2 matrix, we only need the first two terms of the state transition matrix to find the coefficients a_i, that is,

$$e^{At} = a_0 I + a_1 A \tag{5.100}$$

The constants a_0 and a_1 are found from

$$\begin{aligned} a_0 + a_1\lambda_1 &= e^{\lambda_1 t} \\ a_0 + a_1\lambda_2 &= e^{\lambda_2 t} \end{aligned} \tag{5.101}$$

and with $\lambda_1 = 1$ and $\lambda_2 = -1$, we obtain

$$\begin{aligned} a_0 + a_1 &= e^{t} \\ a_0 - a_1 &= e^{-t} \end{aligned} \tag{5.102}$$

and simultaneous solution of (5.102) yields

$$a_0 = \frac{e^t + e^{-t}}{2} = \cosh t$$

$$a_1 = \frac{e^t - e^{-t}}{2} = \sinh t$$

By substitution of these values into (5.100), we obtain

$$e^{At} = \cosh t\,\mathrm{I} + \sinh t\,\mathrm{A} = \cosh t \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \sinh t \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} \cosh t + \sinh t & 0 \\ \sinh t & \cosh t - \sinh t \end{bmatrix} \tag{5.103}$$

The values of the vector $\mathbf{x}$ are found from

$$x(t) = e^{A(t-t_0)}x_0 + e^{At}\int_{t_0}^{t} e^{-A\tau}bu(\tau)d\tau = e^{At}x_0 + e^{At}\int_{0}^{t} e^{-A\tau}b\delta(\tau)d\tau \tag{5.104}$$

Using the sifting property of the delta function we find that (5.104) reduces to

$$x(t) = e^{At}x_0 + e^{At}b = e^{At}(x_0 + b) = e^{At}\left\{\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right\} = e^{At}\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cosh t + \sinh t & 0 \\ \sinh t & \cosh t - \sinh t \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Therefore,

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \cosh t - \sinh t \end{bmatrix} = \begin{bmatrix} 0 \\ e^{-t} \end{bmatrix} \tag{5.105}$$

5.7 Relationship between State Equations and Laplace Transform

In this section, we will show that the state transition matrix can be computed from the Inverse Laplace transform. We will also show that the transfer function can be found from the coefficient matrices of the state equations.

Consider the state equation

$$\dot{x} = Ax + bu \tag{5.106}$$

Taking the Laplace of both sides of (5.106), we obtain

$$sX(s) - x(0) = AX(s) + bU(s)$$

or

$$(sI - A)X(s) = x(0) + bU(s) \tag{5.107}$$

Multiplying both sides of (5.107) by $(sI - A)^{-1}$, we obtain

$$X(s) = (sI - A)^{-1}x(0) + (sI - A)^{-1}bU(s) \tag{5.108}$$

Comparing (5.108) with

$$x(t) = e^{At}x_0 + e^{At}\int_0^t e^{-A\tau}bu(\tau)d\tau \tag{5.109}$$

we observe that the right side of (5.108) is the Laplace transform of (5.109). Therefore, we can compute the state transition matrix e^{At} from the Inverse Laplace of $(sI - A)^{-1}$, that is, we can use the relation

$$\boxed{e^{At} = \mathscr{L}^{-1}\{(sI - A)^{-1}\}} \tag{5.110}$$

Next, we consider the output state equation

$$y = Cx + du \tag{5.111}$$

Taking the Laplace of both sides of (5.111), we obtain

$$Y(s) = CX(s) + dU(s) \tag{5.112}$$

and using (5.108), we obtain

$$Y(s) = C(sI - A)^{-1}x(0) + [C(sI - A)^{-1}b + d]U(s) \tag{5.113}$$

If the initial condition $x(0) = 0$, (5.113) reduces to

$$Y(s) = [C(sI - A)^{-1}b + d]U(s) \tag{5.114}$$

In (5.114), U(s) is the Laplace transform of the input u(t); then, division of both sides by U(s) yields the transfer function

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}b + d \tag{5.115}$$

Example 5.12

In the circuit of Figure 5.11, all initial conditions are zero. Compute the state transition matrix e^{At} using the Inverse Laplace transform method.

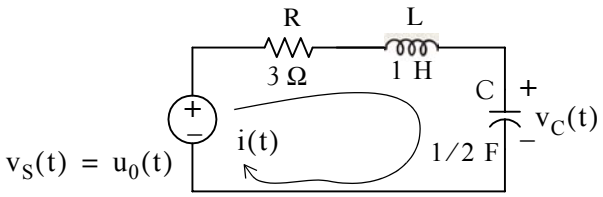


Figure 5.11. Circuit for Example 5.12

Solution:

For this circuit,

$$i = i_L$$

and

$$Ri_L + L\frac{di_L}{dt} + v_C = u_0(t)$$

Substitution of given values and rearranging, yields

$$\frac{di_L}{dt} = -3i_L - v_C + 1 \tag{5.116}$$

Now, we define the state variables

$$x_1 = i_L$$

and

$$x_2 = v_C$$

Then,

$$\dot{x}_1 = \frac{di_L}{dt} = -3i_L - v_C + 1 \tag{5.117}$$

and

$$\dot{x}_2 = \frac{dv_C}{dt}$$

Also,

$$i_L = C\frac{dv_C}{dt} = 0.5\frac{dv_C}{dt} \tag{5.118}$$

and thus,

$$x_1 = i_L = 0.5\frac{dv_C}{dt} = 0.5\dot{x}_2$$

or

$$\dot{x}_2 = 2x_1 \tag{5.119}$$

Therefore, from (5.117) and (5.119) we obtain the state equations

$$\begin{aligned}\dot{x}_1 &= -3x_1 - x_2 + 1\\ \dot{x}_2 &= 2x_1\end{aligned} \tag{5.120}$$

and in matrix form,

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-3 & -1\\ 2 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix}1 \tag{5.121}$$

By inspection,

$$A = \begin{bmatrix}-3 & -1\\ 2 & 0\end{bmatrix} \tag{5.122}$$

Now, we will find the state transition matrix from

$$e^{At} = \mathcal{L}^{-1}\{(sI - A)^{-1}\} \tag{5.123}$$

where

$$(sI - A) = \begin{bmatrix}s & 0\\ 0 & s\end{bmatrix} - \begin{bmatrix}-3 & -1\\ 2 & 0\end{bmatrix} = \begin{bmatrix}s+3 & 1\\ -2 & s\end{bmatrix}$$

Then,

$$(sI - A)^{-1} = \frac{\text{adj}(sI - A)}{\det(sI - A)} = \frac{1}{s^2 + 3s + 2}\begin{bmatrix}s & -1\\ 2 & s+3\end{bmatrix} = \begin{bmatrix}\dfrac{s}{(s+1)(s+2)} & \dfrac{-1}{(s+1)(s+2)}\\ \dfrac{2}{(s+1)(s+2)} & \dfrac{s+3}{(s+1)(s+2)}\end{bmatrix}$$

We find the Inverse Laplace of each term by partial fraction expansion. Thus,

$$e^{At} = \mathcal{L}^{-1}\{(sI - A)^{-1}\} = \begin{bmatrix}-e^{-t} + 2e^{-2t} & -e^{-t} + e^{-2t}\\ 2e^{-t} - 2e^{-2t} & 2e^{-t} - e^{-2t}\end{bmatrix}$$

Now, we can find the state variables representing the inductor current and the capacitor voltage from

$$x(t) = e^{At}x_0 + e^{At}\int_0^t e^{-A\tau}bu(\tau)d\tau$$

using the procedure of Example 5.11.

MATLAB provides two very useful functions to convert state–space (state equations), to transfer function (s–domain), and vice versa. The function **ss2tf** (state–space to transfer function) converts the state space equations

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned} \quad * \qquad (5.124)$$

to the rational transfer function form

$$G(s) = \frac{N(s)}{D(s)} \qquad (5.125)$$

This is used with the statement **[num,den]=ss2tf(A,B,C,D,iu)** where **A, B, C, D** are the matrices of (5.124) and **iu** is 1 if there is only one input. The MATLAB **help** command provides the following information:

help ss2tf

```
 SS2TF  State-space to transfer function conversion.
    [NUM,DEN] = SS2TF(A,B,C,D,iu) calculates the
     transfer function:
                 NUM(s)           -1
        G(s) = -------- = C(sI-A) B + D
                 DEN(s)
    of the system:
        x = Ax + Bu
        y = Cx + Du
from the iu'th input. Vector DEN contains the coefficients of   the
denominator in descending powers of s. The numerator coefficients are
returned in matrix NUM with as many rows as there     are outputs y.

    See also TF2SS
```

The other function, **tf2ss**, converts the transfer function of (5.125) to the state–space equations of (5.124). It is used with the statement **[A,B,C,D]=tf2ss(num,den)** where **A, B, C**, and **D** are the matrices of (5.124), and num, den are N(s) and D(s) of (5.125) respectively. The MATLAB **help** command provides the following information:

* *We have used capital letters for vectors b and c to be consistent with MATLAB's designations.*

help tf2ss

```
 TF2SS  Transfer function to state-space conversion.
    [A,B,C,D] = TF2SS(NUM,DEN) calculates the state-space
    representation:

        x = Ax + Bu
        y = Cx + Du
    of the system:
                   NUM(s)
        G(s) = --------
                   DEN(s)
from a single input. Vector DEN must contain the coefficients of the
denominator in descending powers of s. Matrix NUM must contain the
numerator coefficients with as many rows as there are outputs y. The
A,B,C,D matrices are returned in controller canonical form. This calcu-
lation also works for discrete systems. To avoid confusion when using
this function with discrete systems, always use a numerator polynomial
that has been padded with zeros to make it the same length as the denom-
inator. See the User's guide for more details.

    See also SS2TF.
```

Example 5.13

For the circuit of Figure 5.12, all initial conditions are zero.

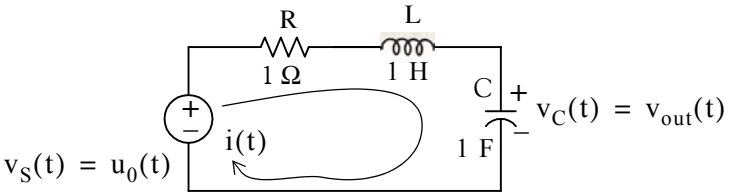


Figure 5.12. Circuit for Example 5.13

a. Derive the state equations and express them in matrix form as

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

b. Derive the transfer function

$$G(s) = \frac{N(s)}{D(s)}$$

c. Verify your answers with MATLAB.

Solution:

a. The differential equation describing the circuit is

$$Ri + L\frac{di}{dt} + v_C = u_0(t)$$

and with the given values,

$$i + \frac{di}{dt} + v_C = u_0(t)$$

or

$$\frac{di}{dt} = -i - v_C + u_0(t)$$

We let

$$x_1 = i_L = i$$

and

$$x_2 = v_C = v_{out}$$

Then,

$$\dot{x}_1 = \frac{di}{dt}$$

and

$$\dot{x}_2 = \frac{dv_c}{dt} = x_1$$

Thus, the state equations are

$$\begin{aligned}\dot{x}_1 &= -x_1 - x_2 + u_0(t)\\ \dot{x}_2 &= x_1\\ y &= x_2\end{aligned}$$

and in matrix form,

$$\begin{aligned}\dot{x} = Ax + Bu \leftrightarrow \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} &= \begin{bmatrix}-1 & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix}u_0(t)\\ y = Cx + Du \leftrightarrow y &= \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\end{bmatrix}u_0(t)\end{aligned} \tag{5.126}$$

b. The s – domain circuit is shown in Figure 5.13 below.

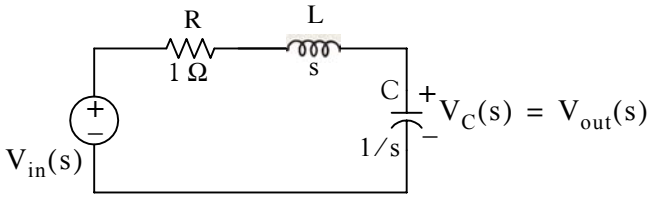


Figure 5.13. Transformed circuit for Example 5.13

By the voltage division expression,

$$V_{out}(s) = \frac{1/s}{1 + s + 1/s} V_{in}(s)$$

or

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{s^2 + s + 1}$$

Therefore,

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{s^2 + s + 1} \tag{5.127}$$

c.

```
A = [−1  −1; 1  0]; B = [1  0]';  C = [0  1]; D = [0];      % The matrices of (5.126)
[num, den] = ss2tf(A, B, C, D, 1)                           % Verify coefficients of G(s) in (5.127)

num =
   0      0      1
den =
  1.0000    1.0000    1.0000

num = [0  0  1]; den = [1  1  1];                           % The coefficients of G(s) in (5.127)
[A  B  C  D] = tf2ss(num, den)                              % Verify the matrices of (5.126)

A =
  -1    -1
   1     0
B =
   1
   0
C =
   0     1
D =
   0
```

The equivalence between the state–space equations of (5.126) and the transfer function of (5.127) is also evident from the Simulink models shown in Figure 5.14 where for the **State–Space** block Function Block Parameters dialog box we have entered:

A: [–1 –1; 3/4 0]
B: [1 0]'
C: [0 1]
D: [0]
Initial conditions: [0 0]

For the **Transfer Fcn** block Function Block Parameters dialog box we have entered:

Numerator coefficient: [1]
Denominator coefficient: [1 1 1]

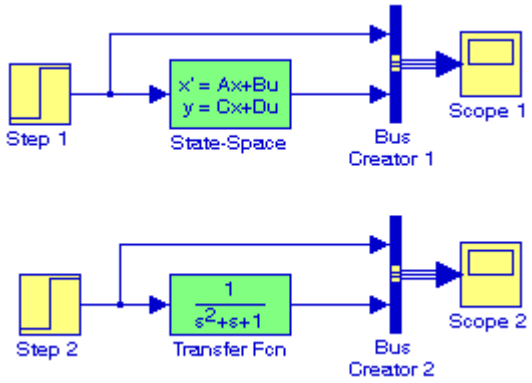


Figure 5.14. Models to show the equivalence between relations (5.126) and (5.127)

After the simulation command is executed, both Scope 1 and Scope 2 blocks display the input and output waveforms shown in Figure 5.15.

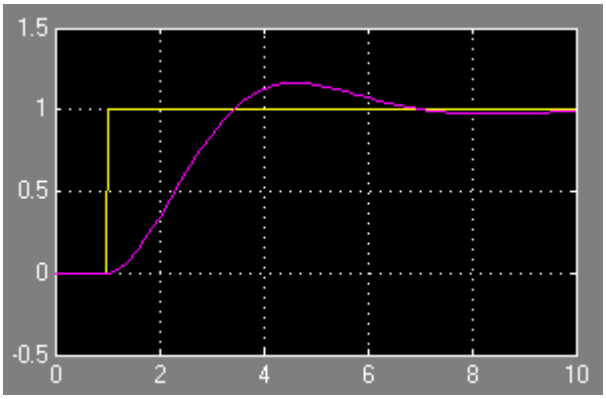


Figure 5.15. Waveforms displayed by Scope 1 and Scope 2 blocks for the models in Figure 5.14

5.8 Summary

- An nth–order differential equation can be resolved to n first–order simultaneous differential equations with a set of auxiliary variables called state variables. The resulting first–order differential equations are called state–space equations, or simply state equations.
- The state–space equations can be obtained either from the nth–order differential equation, or directly from the network, provided that the state variables are chosen appropriately.
- When we obtain the state equations directly from given circuits, we choose the state variables to represent inductor currents and capacitor voltages.
- The state variable method offers the advantage that it can also be used with non–linear and time–varying devices.
- If a circuit contains only one energy–storing device, the state equations are written as

$$\dot{x} = \alpha x + \beta u$$
$$y = k_1 x + k_2 u$$

where α, β, k_1, and k_2 are scalar constants, and the initial condition, if non–zero, is denoted as

$$x_0 = x(t_0)$$

- If α and β are scalar constants, the solution of $\dot{x} = \alpha x + \beta u$ with initial condition $x_0 = x(t_0)$ is obtained from the relation

$$x(t) = e^{\alpha(t-t_0)}x_0 + e^{\alpha t}\int_{t_0}^{t} e^{-\alpha\tau}\beta u(\tau)d\tau$$

- The solution of the state equations pair

$$\dot{x} = Ax + bu$$
$$y = Cx + du$$

where A and C are 2×2 or higher order matrices, and b and d are column vectors with two or more rows, entails the computation of the state transition matrix e^{At}, and integration of

$$x(t) = e^{A(t-t_0)}x_0 + e^{At}\int_{t_0}^{t} e^{-A\tau}bu(\tau)d\tau$$

- The eigenvalues λ_i, where $i = 1, 2, \ldots, n$, of an $n\times n$ matrix A are the roots of the nth order polynomial

$$\det[A - \lambda I] = 0$$

where I is the $n\times n$ identity matrix.

- The Cayley–Hamilton theorem states that a matrix can be expressed as an $(n-1)$th degree polynomial in terms of the matrix A as

$$e^{At} = a_0 I + a_1 A + a_2 A^2 + \ldots + a_{n-1} A^{n-1}$$

where the coefficients a_i are functions of the eigenvalues λ.

- If all eigenvalues of a given matrix A are distinct, that is, if

$$\lambda_1 \neq \lambda_2 \neq \lambda_3 \neq \ldots \neq \lambda_n$$

the coefficients a_i are found from the simultaneous solution of the system of equations

$$\begin{aligned}
a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1} &= e^{\lambda_1 t} \\
a_0 + a_1\lambda_2 + a_2\lambda_2^2 + \ldots + a_{n-1}\lambda_2^{n-1} &= e^{\lambda_2 t} \\
&\ldots \\
a_0 + a_1\lambda_n + a_2\lambda_n^2 + \ldots + a_{n-1}\lambda_n^{n-1} &= e^{\lambda_n t}
\end{aligned}$$

- If some or all eigenvalues of matrix A are repeated, that is, if

$$\lambda_1 = \lambda_2 = \lambda_3 \ldots = \lambda_m,\ \lambda_{m+1},\ \lambda_n$$

the coefficients a_i of the state transition matrix are found from the simultaneous solution of the system of equations

$$\begin{aligned}
a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1} &= e^{\lambda_1 t} \\
\frac{d}{d\lambda_1}(a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1}) &= \frac{d}{d\lambda_1}e^{\lambda_1 t} \\
\frac{d^2}{d\lambda_1^2}(a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1}) &= \frac{d^2}{d\lambda_1^2}e^{\lambda_1 t} \\
&\ldots \\
\frac{d^{m-1}}{d\lambda_1^{m-1}}(a_0 + a_1\lambda_1 + a_2\lambda_1^2 + \ldots + a_{n-1}\lambda_1^{n-1}) &= \frac{d^{m-1}}{d\lambda_1^{m-1}}e^{\lambda_1 t} \\
a_0 + a_1\lambda_{m+1} + a_2\lambda_{m+1}^2 + \ldots + a_{n-1}\lambda_{m+1}^{n-1} &= e^{\lambda_{m+1} t} \\
&\ldots \\
a_0 + a_1\lambda_n + a_2\lambda_n^2 + \ldots + a_{n-1}\lambda_n^{n-1} &= e^{\lambda_n t}
\end{aligned}$$

- We can use the MATLAB **eig(x)** function to find the eigenvalues of an $n \times n$ matrix.
- A column vector X that satisfies the relation

$$AX = \lambda X$$

where A is an $n \times n$ matrix and λ is a scalar number, is called an eigenvector.

- There is a different eigenvector for each eigenvalue.
- Eigenvectors are generally expressed as unit eigenvectors, that is, they are normalized to unit length. This is done by dividing each component of the eigenvector by the square root of the sum of the squares of their components, so that the sum of the squares of their components is equal to unity.
- Two vectors X and Y are said to be orthogonal if their inner (dot) product is zero.
- A set of eigenvectors constitutes an orthonormal basis if the set is normalized (expressed as unit eigenvectors) and these vector are mutually orthogonal.
- The state transition matrix can be computed from the Inverse Laplace transform using the relation

$$e^{At} = \mathcal{L}^{-1}\{(sI - A)^{-1}\}$$

- If $U(s)$ is the Laplace transform of the input $u(t)$ and $Y(s)$ is the Laplace transform of the output $y(t)$, the transfer function can be computed using the relation

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}b + d$$

- MATLAB provides two very useful functions to convert state–space (state equations), to transfer function (s-domain), and vice versa. The function **ss2tf** (state–space to transfer function) converts the state space equations to the transfer function equivalent, and the function **tf2ss**, converts the transfer function to state–space equations.

5.9 Exercises

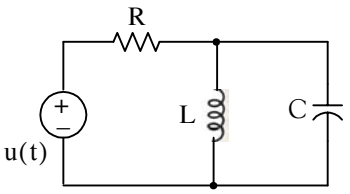
1. Express the integrodifferential equation below as a matrix of state equations where k_1, k_2, and k_3 are constants.

$$\frac{dv^2}{dt^2} + k_3\frac{dv}{dt} + k_2 v + k_1\int_0^t v\,dt = \sin 3t + \cos 3t$$

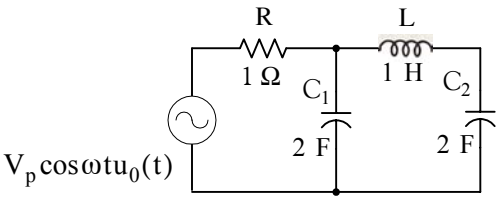
2. Express the matrix of the state equations below as a single differential equation, and let $x(y) = y(t)$.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -2 & -3 & -4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

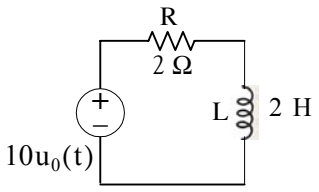
3. For the circuit below, all initial conditions are zero, and $u(t)$ is any input. Write state equations in matrix form.



4. In the circuit below, all initial conditions are zero. Write state equations in matrix form.



5. In the below, $i_L(0^-) = 2$ A. Use the state variable method to find $i_L(t)$ for $t > 0$.



6. Compute the eigenvalues of the matrices A, B, and C below.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} a & 0 \\ -a & b \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}$$

Hint: One of the eigenvalues of matrix C is -1.

7. Compute e^{At} given that

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}$$

Observe that this is the same matrix as C of Exercise 6.

8. Find the solution of the matrix state equation $\dot{x} = Ax + bu$ given that

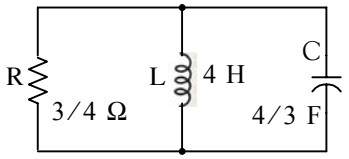
$$A = \begin{bmatrix} 1 & 0 \\ -2 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_0 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad u = \delta(t), \quad t_0 = 0$$

9. In the circuit below, $i_L(0^-) = 0$, and $v_C(0^-) = 1$ V.

a. Write state equations in matrix form.

b. Compute e^{At} using the Inverse Laplace transform method.

c. Find $i_L(t)$ and $v_C(t)$ for $t > 0$.



5.10 Solutions to End–of–Chapter Exercises

1. Differentiating the given integrodifferential equation with respect to t we obtain

$$\frac{dv^3}{dt^3}+k_3\frac{dv^2}{dt^2}+k_2\frac{dv}{dt}+k_1v = 3\cos 3t-3\sin 3t = 3(\cos 3t-\sin 3t)$$

or

$$\frac{dv^3}{dt^3}=-k_3\frac{dv^2}{dt^2}-k_2\frac{dv}{dt}-k_1v+3(\cos 3t-\sin 3t) \quad (1)$$

We let

$$v = x_1 \qquad \frac{dv}{dt} = x_2 = \dot{x}_1 \qquad \frac{dv^2}{dt^2} = x_3 = \dot{x}_2$$

Then,

$$\frac{dv^3}{dt^3} = \dot{x}_3$$

and by substitution into (1)

$$\dot{x}_3 = -k_1x_1-k_2x_2-k_3x_3+3(\cos 3t-\sin 3t)$$

and thus the state equations are

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = x_3$$
$$\dot{x}_3 = -k_1x_1-k_2x_2-k_3x_3+3(\cos 3t-\sin 3t)$$

and in matrix form

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -k_1 & -k_2 & -k_3\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}\cdot 3(\cos 3t-\sin 3t)$$

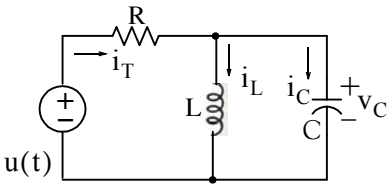
2. Expansion of the given matrix yields

$$\dot{x}_1 = x_2 \quad \dot{x}_2 = x_3 \quad \dot{x}_3 = x_2 \quad \dot{x}_4 = -x_1-2x_2-3x_3-4x_4+u(t)$$

Letting x = y we obtain

$$\frac{dy^4}{dt^4}+4\frac{dy^3}{dt^3}+3\frac{dy^2}{dt^2}+2\frac{dy}{dt}+y = u(t)$$

3.



We let $i_L = x_1$ and $v_C = x_2$. By KCL, $i_T = i_L + i_C$ or

$$\frac{u(t) - v_C}{R} = i_L + C\frac{dv_C}{dt}$$

or

$$\frac{u(t) - x_2}{R} = x_1 + C\dot{x}_2$$

Also,

$$x_2 = L\dot{x}_1$$

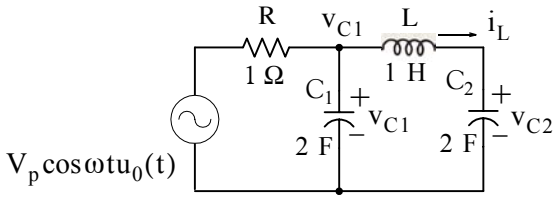
Then,

$$\dot{x}_1 = \frac{1}{L}x_2 \text{ and } \dot{x}_2 = -\frac{1}{C}x_1 - \frac{1}{RC}x_2 + \frac{1}{RC}u(t)$$

and in matrix form

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1/L\\ -1/C & -1/RC\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1/RC\end{bmatrix}\cdot u(t)$$

4.



We let $i_L = x_1$, $v_{C1} = x_2$, and $v_{C2} = x_3$. By KCL,

$$\frac{v_{C1} - V_p\cos\omega t u_0(t)}{1} + 2\frac{dv_{C1}}{dt} + i_L = 0$$

or

$$x_2 - V_p\cos\omega t u_0(t) + 2\dot{x}_2 + x_1 = 0$$

or

$$\dot{x}_2 = -\frac{1}{2}x_1 - \frac{1}{2}x_2 + \frac{1}{2}V_p\cos\omega t u_0(t) \quad (1)$$

By KVL,

$$v_{C1} = L\frac{di_L}{dt} + v_{C2}$$

or

$$x_2 = 1\dot{x}_1 + x_3$$

or

$$\dot{x}_1 = x_2 - x_3 \quad (2)$$

Also,

$$i_L = C\frac{dv_{C2}}{dt}$$

or

$$x_1 = 2\dot{x}_3$$

or

$$\dot{x}_3 = \frac{1}{2}x_1 \quad (3)$$

Combining (1), (2), and (3) into matrix form we obtain

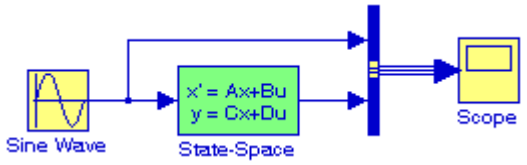
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 \\ -1/2 & -1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/2 \\ 0 \end{bmatrix} \cdot V_p \cos\omega t u_0(t)$$

We will create a Simulink model with $V_p = 1$ and output $y = x_3$. The model is shown below where for the State–Space block Function Block Parameters dialog box we have entered:

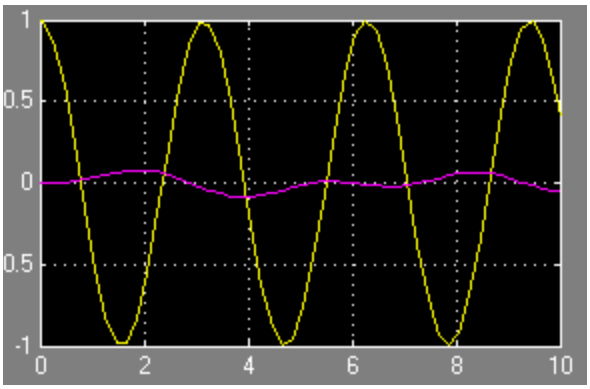
A: [0 1 –1; –1/2 –1/2 0; 1/2 0 0]
B: [0 1/2 0]'
C: [0 0 1]
D: [0]
Initial conditions: [0 0 0]

and for the Sine Wave block Function Block Parameters dialog box we have entered:

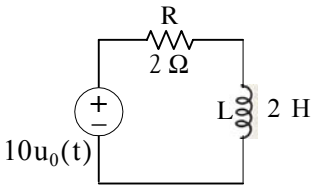
Amplitude: 1
Phase: pi/2



The input and output waveforms are shown below.



5.



From (5.21) of Example 5.4, Page 5–6,

$$\dot{x} = -\frac{R}{L}x + \frac{1}{L}v_S u_0(t)$$

For this exercise, $\alpha = -R/L = -1$ and $b = 10 \times (1/L) = 5$. Then,

$$x(t) = e^{\alpha(t-t_0)}x_0 + e^{\alpha t}\int_{t_0}^{t} e^{-\alpha\tau}\beta u(\tau)d\tau$$

$$= e^{-1(t-0)}2 + e^{-t}\int_0^t e^{\tau}5u_0(\tau)d\tau = 2e^{-t} + 5e^{-t}\int_0^t e^{\tau}d\tau$$

$$= 2e^{-t} + 5e^{-t}(e^{t} - 1) = 2e^{-t} + 5 - 5e^{-t} = (5 - 3e^{-t})u_0(t)$$

and denoting the current i_L as the output y we obtain

$$y(t) = x(t) = (5 - 3e^{-t})u_0(t)$$

6.

a.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & -1 \end{bmatrix} \qquad \det(A - \lambda I) = \det\left(\begin{bmatrix} 1 & 2 \\ 3 & -1 \end{bmatrix} - \lambda\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = \det\begin{bmatrix} 1-\lambda & 2 \\ 3 & -1-\lambda \end{bmatrix} = 0$$

$$(1 - \lambda)(-1 - \lambda) - 6 = 0$$

$$-1-\lambda+\lambda+\lambda^2-6=0$$

$$\lambda^2=7$$

and thus

$$\lambda_1=\sqrt{7}\qquad\lambda_2=-\sqrt{7}$$

b.

$$B=\begin{bmatrix}a&0\\-a&b\end{bmatrix}\qquad \det(B-\lambda I)=\det\left(\begin{bmatrix}a&0\\-a&b\end{bmatrix}-\lambda\begin{bmatrix}1&0\\0&1\end{bmatrix}\right)=\det\begin{bmatrix}a-\lambda&0\\-a&b-\lambda\end{bmatrix}=0$$

$$(a-\lambda)(b-\lambda)=0$$

and thus

$$\lambda_1=a\qquad\lambda_2=b$$

c.

$$C=\begin{bmatrix}0&1&0\\0&0&1\\-6&-11&-6\end{bmatrix}\qquad \det(C-\lambda I)=\det\left(\begin{bmatrix}0&1&0\\0&0&1\\-6&-11&-6\end{bmatrix}-\lambda\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\right)$$

$$=\det\begin{bmatrix}-\lambda&1&0\\0&-\lambda&1\\-6&-11&-6-\lambda\end{bmatrix}=0$$

$$\lambda^2(-6-\lambda)-6-(-11)(-\lambda)=\lambda^3+6\lambda^2+11\lambda+6=0$$

and it is given that $\lambda_1=-1$. Then,

$$\frac{\lambda^3+6\lambda^2+11\lambda+6}{(\lambda+1)}=\lambda^2+5\lambda+6\Rightarrow(\lambda+1)(\lambda+2)(\lambda+3)=0$$

and thus

$$\lambda_1=-1\qquad\lambda_2=-2\qquad\lambda_1=-3$$

7.

a. Matrix A is the same as Matrix C in Exercise 6. Then,

$$\lambda_1=-1\qquad\lambda_2=-2\qquad\lambda_1=-3$$

and since A is a 3×3 matrix the state transition matrix is

$$e^{At}=a_0I+a_1A+a_2A^2\quad(1)$$

Then,

$$a_0 + a_1\lambda_1 + a_2\lambda_1^2 = e^{\lambda_1 t} \Rightarrow a_0 - a_1 + a_2 = e^{-t}$$

$$a_0 + a_1\lambda_2 + a_2\lambda_2^2 = e^{\lambda_2 t} \Rightarrow a_0 - 2a_1 + 4a_2 = e^{-2t}$$

$$a_0 + a_1\lambda_3 + a_2\lambda_3^2 = e^{\lambda_3 t} \Rightarrow a_0 - 3a_1 + 9a_2 = e^{-3t}$$

```
syms t; A=[1 –1 1; 1 –2 4; 1 –3 9];...
a=sym('[exp(–t); exp(–2*t); exp(–3*t)]'); x=A\a; fprintf(' \n');...
disp('a0 = '); disp(x(1)); disp('a1 = '); disp(x(2)); disp('a2 = '); disp(x(3))
```

```
a0 =
3*exp(-t)-3*exp(-2*t)+exp(-3*t)
a1 =
5/2*exp(-t)-4*exp(-2*t)+3/2*exp(-3*t)
a2 =
1/2*exp(-t)-exp(-2*t)+1/2*exp(-3*t)
```

Thus,

$$a_0 = 3e^{-t} - 3e^{-2t} + 3e^{-3t}$$

$$a_1 = 2.5e^{-t} - 4e^{-2t} + 1.5e^{-3t}$$

$$a_2 = 0.5e^{-t} - e^{-2t} + 0.5e^{-3t}$$

Now, we compute e^{At} of (1) with the following MATLAB script:

```
syms t; a0=3*exp(–t)–3*exp(–2*t)+exp(–3*t); a1=5/2*exp(–t)–4*exp(–2*t)+3/2*exp(–3*t);...
a2=1/2*exp(–t)-exp(–2*t)+1/2*exp(–3*t); A=[0 1 0; 0 0 1; –6 –11 –6]; fprintf(' \n');...
eAt=a0*eye(3)+a1*A+a2*A^2
```

```
eAt =
[3*exp(-t)-3*exp(-2*t)+exp(-3*t),    5/2*exp(-t)-4*exp(-2*t)+3/
2*exp(-3*t),      1/2*exp(-t)-exp(-2*t)+1/2*exp(-3*t)]
[-3*exp(-t)+6*exp(-2*t)-3*exp(-3*t),   -5/2*exp(-t)+8*exp(-2*t)-
9/2*exp(-3*t),   -1/2*exp(-t)+2*exp(-2*t)-3/2*exp(-3*t)]
[3*exp(-t)-12*exp(-2*t)+9*exp(-3*t), 5/2*exp(-t)-16*exp(-
2*t)+27/2*exp(-3*t),    1/2*exp(-t)-4*exp(-2*t)+9/2*exp(-3*t)]
```

Thus,

$$e^{At} = \begin{bmatrix} 3e^{-t} - 3e^{-2t} + e^{-3t} & 2.5e^{-t} - 4e^{-2t} + 1.5e^{-3t} & 0.5e^{-t} - e^{-2t} + 0.5e^{-3t} \\ -3e^{-t} + 6e^{-2t} - 3e^{-3t} & -2.5e^{-t} + 8e^{-2t} - 4.5e^{-3t} & -0.5e^{-t} + 2e^{-2t} - 1.5e^{-3t} \\ 3e^{-t} - 12e^{-2t} + 9e^{-3t} & 2.5e^{-t} - 16e^{-2t} + 13.5e^{-3t} & 0.5e^{-t} - 4e^{-2t} + 4.5e^{-3t} \end{bmatrix}$$

8.

$$A = \begin{bmatrix} 1 & 0 \\ -2 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_0 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad u = \delta(t), \quad t_0 = 0$$

$$\begin{aligned} x(t) &= e^{A(t-0)}x_0 + e^{At}\int_0^t e^{-A\tau}bu(\tau)d\tau = e^{At}x_0 + e^{At}\int_0^t e^{-A\tau}b\delta(\tau)d\tau \\ &= e^{At}x_0 + e^{At}b = e^{At}(x_0 + b) = e^{At}\left(\begin{bmatrix} -1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) = e^{At}\begin{bmatrix} 0 \\ 2 \end{bmatrix} \end{aligned} \tag{1}$$

We use the following MATLAB script to find the eigenvalues λ_1 and λ_2.

```
A=[1  0; –2  2]; lambda=eig(A); fprintf(' \n');...
fprintf('lambda1 = %4.2f  \t',lambda(1)); fprintf('lambda2 = %4.2f  \t',lambda(2))
```

```
lambda1 = 2.00   lambda2 = 1.00
```

Next,

$$a_0 + a_1\lambda_1 = e^{\lambda_1 t} \Rightarrow a_0 + a_1 = e^t$$

$$a_0 + a_1\lambda_2 = e^{\lambda_2 t} \Rightarrow a_0 + 2a_1 = e^{2t}$$

Then,

$$a_0 = 2e^t - e^{2t} \qquad a_1 = e^{2t} - e^t$$

and

$$\begin{aligned} e^{At} &= a_0 I + a_1 A = (2e^t - e^{2t})\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + (e^{2t} - e^t)\begin{bmatrix} 1 & 0 \\ -2 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 2e^t - e^{2t} & 0 \\ 0 & 2e^t - e^{2t} \end{bmatrix} + \begin{bmatrix} e^{2t} - e^t & 0 \\ -2e^{2t} + 2e^t & 2e^{2t} - 2e^t \end{bmatrix} = \begin{bmatrix} e^t & 0 \\ 2e^t - 2e^{2t} & e^{2t} \end{bmatrix} \end{aligned}$$

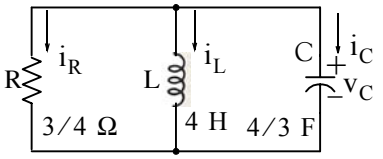
By substitution into (1) we obtain

$$x(t) = e^{At}\begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} e^t & 0 \\ 2e^t - 2e^{2t} & e^{2t} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2e^{2t} \end{bmatrix}$$

and thus

$$x_1 = 0 \qquad x_2 = 2e^{2t}$$

9.



$$i_L(0^-) = 0$$
$$v_C(0^-) = 1 \text{ V}$$

We let

$$x_1 = i_L \qquad x_2 = v_C$$

Then,

a.

$$i_R + i_L + i_C = 0$$

$$\frac{v_C}{R} + i_L + C\frac{v_C}{dt} = 0$$

$$\frac{x_2}{3/4} + x_1 + \frac{4}{3}\dot{x}_2 = 0$$

or

$$\dot{x}_2 = -\frac{3}{4}x_1 - x_2 \quad (1)$$

Also,

$$v_L = v_C = L\frac{di_L}{dt} = 4\dot{x}_1 = x_2$$

or

$$\dot{x}_1 = \frac{1}{4}x_2 \quad (2)$$

From (1) and (2)

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1/4\\ -3/4 & -1\end{bmatrix} \cdot \begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

and thus

$$A = \begin{bmatrix}0 & 1/4\\ -3/4 & -1\end{bmatrix}$$

b.

$$e^{At} = \mathcal{L}^{-1}\{[sI - A]^{-1}\}$$

$$[sI-A]=\begin{bmatrix}s & 0\\ 0 & s\end{bmatrix}-\begin{bmatrix}0 & 1/4\\ -3/4 & -1\end{bmatrix}=\begin{bmatrix}s & -1/4\\ 3/4 & s+1\end{bmatrix}$$

$$\Delta=\det[sI-A]=\det\begin{bmatrix}s & -1/4\\ 3/4 & s+1\end{bmatrix}=s^2+s+3/16=(s+1/4)(s+3/4)$$

$$\text{adj}[sI-A]=\text{adj}\begin{bmatrix}s & -1/4\\ 3/4 & s+1\end{bmatrix}=\begin{bmatrix}s+1 & 1/4\\ -3/4 & s\end{bmatrix}$$

$$[sI-A]^{-1}=\frac{1}{\Delta}\text{adj}[sI-A]=\frac{1}{(s+1/4)(s+3/4)}\begin{bmatrix}s+1 & 1/4\\ -3/4 & s\end{bmatrix}$$

$$=\begin{bmatrix}\dfrac{s+1}{(s+1/4)(s+3/4)} & \dfrac{1/4}{(s+1/4)(s+3/4)}\\ \dfrac{-3/4}{(s+1/4)(s+3/4)} & \dfrac{s}{(s+1/4)(s+3/4)}\end{bmatrix}$$

We use MATLAB to find $e^{At}=\mathcal{L}^{-1}\{[sI-A]^{-1}\}$ with the script below.

```
syms s t

Fs1=(s+1)/(s^2+s+3/16); Fs2=(1/4)/(s^2+s+3/16); Fs3=(–3/4)/(s^2+s+3/16); Fs4=s/
(s^2+s+3/16);...
fprintf(' \n'); disp('a11 = '); disp(simple(ilaplace(Fs1))); disp('a12 = '); disp(simple(ila-
place(Fs2)));...
disp('a21 = '); disp(simple(ilaplace(Fs3))); disp('a22 = '); disp(simple(ilaplace(Fs4)))
```

```
a11 =
-1/2*exp(-3/4*t)+3/2*exp(-1/4*t)
a12 =
1/2*exp(-1/4*t)-1/2*exp(-3/4*t)
a21 =
-3/2*exp(-1/4*t)+3/2*exp(-3/4*t)
a22 =
3/2*exp(-3/4*t)-1/2*exp(-1/4*t)
```

Thus,

$$e^{At}=\begin{bmatrix}1.5e^{-0.25t}-0.5e^{-0.75t} & 0.5e^{-0.25t}-0.5e^{-0.75t}\\ -1.5e^{-0.25t}+1.5e^{-0.75t} & -0.5e^{-0.25t}+1.5e^{-0.75t}\end{bmatrix}$$

c.

$$x(t) = e^{A(t-0)}x_0 + e^{At}\int_0^t e^{-A\tau}bu(\tau)d\tau = e^{At}x_0 + 0 = e^{At}\left(\begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}\right)$$

$$= \begin{bmatrix}1.5e^{-0.25t} - 0.5e^{-0.75t} & 0.5e^{-0.25t} - 0.5e^{-0.75t}\\ -1.5e^{-0.25t} + 1.5e^{-0.75t} & -0.5e^{-0.25t} + 1.5e^{-0.75t}\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0.5e^{-0.25t} - 0.5e^{-0.75t}\\ -0.5e^{-0.25t} + 1.5e^{-0.75t}\end{bmatrix}$$

and thus for $t > 0$,

$$x_1 = i_L = 0.5e^{-0.25t} - 0.5e^{-0.75t} \qquad x_2 = v_C = -0.5e^{-0.25t} + 1.5e^{-0.75t}$$

Chapter 6

The Impulse Response and Convolution

This chapter begins with the definition of the impulse response, that is, the response of a circuit that is subjected to the excitation of the impulse function. Then, it defines convolution and how it is applied to circuit analysis. Evaluation of the convolution integral using graphical methods is also presented and illustrated with several examples.

6.1 The Impulse Response in Time Domain

In this section we will discuss the impulse response of a network, that is, the output (voltage or current) of a network when the input is the delta function. Of course, the output can be any voltage or current that we choose as the output. The computation of the impulse response assumes zero initial conditions.

We learned in the previous chapter that the state equation

$$\dot{x} = Ax + bu \tag{6.1}$$

has the solution

$$x(t) = e^{A(t-t_0)}x_0 + e^{At}\int_0^t e^{-A\tau}bu(\tau)d\tau \tag{6.2}$$

Therefore, with initial condition $x_0 = 0$, and with the input $u(t) = \delta(t)$, the solution of (6.2) reduces to

$$x(t) = e^{At}\int_0^t e^{-A\tau}b\delta(\tau)d\tau \tag{6.3}$$

Using the sifting property of the delta function, i.e.,

$$\int_{-\infty}^{\infty} f(t)\delta(\tau)d\tau = f(0) \tag{6.4}$$

and denoting the impulse response as $h(t)$, we obtain

$$\boxed{h(t) = e^{At}bu_0(t)} \tag{6.5}$$

where the unit step function $u_0(t)$ is included to indicate that this relation holds for $t > 0$.

Example 6.1

Compute the impulse response of the series RC circuit of Figure 6.1 in terms of the constants R and C, where the response is considered to be the voltage across the capacitor, and $v_C(0^-) = 0$. Then, compute the current through the capacitor.

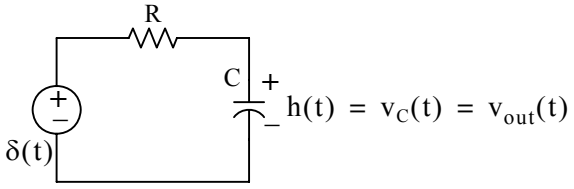


Figure 6.1. Circuit for Example 6.1

Solution:

We assign currents i_C and i_R with the directions shown in Figure 6.2, and we apply KCL.

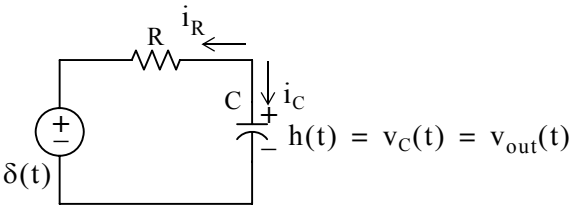


Figure 6.2. Application of KCL for the circuit for Example 6.1

Then,

$$i_R + i_C = 0$$

or

$$C\frac{dv_C}{dt} + \frac{v_C - \delta(t)}{R} = 0 \tag{6.6}$$

We assign the state variable

$$v_C = x$$

Then,

$$\frac{dv_C}{dt} = \dot{x}$$

and (6.6) is written as

$$C\dot{x} + \frac{x}{R} = \frac{\delta(t)}{R}$$

or

$$\dot{x} = -\frac{1}{RC}x + \frac{1}{RC}\delta(t) \tag{6.7}$$

Equation (6.7) has the form

$$\dot{x} = ax + bu$$

and as we found in (6.5),

$$h(t) = e^{At}bu_0(t)$$

For this example,

$$a = -1/RC$$

and

$$b = 1/RC$$

Therefore,

$$h(t) = v_C(t) = e^{-t/RC}\frac{1}{RC}$$

or

$$h(t) = \frac{1}{RC}e^{-t/RC}u_0(t) \tag{6.8}$$

The current i_C can now be computed from

$$i_C = C\frac{dv_C}{dt}$$

Thus,

$$i_C = C\frac{d}{dt}h(t) = C\frac{d}{dt}\left(\frac{1}{RC}e^{-t/RC}u_0(t)\right)$$

$$= -\frac{1}{R^2C}e^{-t/RC} + \frac{1}{R}e^{-t/RC}\delta(t)$$

Using the sampling property of the delta function, we obtain

$$i_C = \frac{1}{R}\delta(t) - \frac{1}{R^2C}e^{-t/RC} \tag{6.9}$$

Example 6.2

For the circuit of Figure 6.3, compute the impulse response $h(t) = v_C(t)$ given that the initial conditions are zero, that is, $i_L(0^-) = 0$, and $v_C(0^-) = 0$.

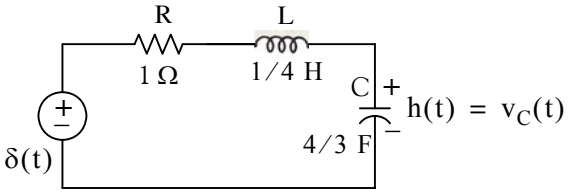


Figure 6.3. Circuit for Example 6.2

Solution:

This is the same circuit as that of Example 5.10, Chapter 5, Page 5–22, where we found that

$$b = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

and

$$e^{At} = \begin{bmatrix} -0.5e^{-t} + 1.5e^{-3t} & -2e^{-t} + 2e^{-3t} \\ \frac{3}{8}e^{-t} - \frac{3}{8}e^{-3t} & 1.5e^{-t} - 0.5e^{-3t} \end{bmatrix}$$

The impulse response is obtained from (6.5), Page 6–1, that is,

$$h(t) = x(t) = e^{At}bu_0(t)$$

then,

$$h(t) = x(t) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -0.5e^{-t} + 1.5e^{-3t} & -2e^{-t} + 2e^{-3t} \\ \frac{3}{8}e^{-t} - \frac{3}{8}e^{-3t} & 1.5e^{-t} - 0.5e^{-3t} \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix} u_0(t) = \begin{bmatrix} -2e^{-t} + 6e^{-3t} \\ \frac{3}{2}e^{-t} - \frac{3}{2}e^{-3t} \end{bmatrix} u_0(t) \quad (6.10)$$

In Example 5.10, Chapter 5, Page 5–22, we defined

$$x_1 = i_L$$

and

$$x_2 = v_C$$

Then,

$$h(t) = x_2 = v_C(t) = 1.5e^{-t} - 1.5e^{-3t}$$

or

$$h(t) = v_C(t) = 1.5(e^{-t} - e^{-3t}) \quad (6.11)$$

Of course, this answer is not the same as that of Example 5.10, because the inputs and initial conditions were defined differently.

6.2 Even and Odd Functions of Time

A function $f(t)$ is an *even function* of time if the following relation holds.

$$\boxed{f(-t) = f(t)} \quad (6.12)$$

that is, if in an even function we replace t with $-t$, the function $f(t)$ does not change. Thus, polynomials with even exponents only, and with or without constants, are even functions. For instance, the cosine function is an even function because it can be written as the power series

$$\cos t = 1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \frac{t^6}{6!} + \ldots$$

Other examples of even functions are shown in Figure 6.4.

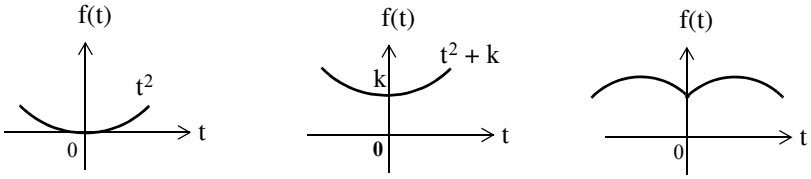


Figure 6.4. Examples of even functions

A function $f(t)$ is an *odd function* of time if the following relation holds.

$$\boxed{-f(-t) = f(t)} \tag{6.13}$$

that is, if in an odd function we replace t with $-t$, we obtain the negative of the function $f(t)$. Thus, polynomials with odd exponents only, and no constants are odd functions. For instance, the sine function is an odd function because it can be written as the power series

$$\sin t = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \ldots$$

Other examples of odd functions are shown in Figure 6.5.

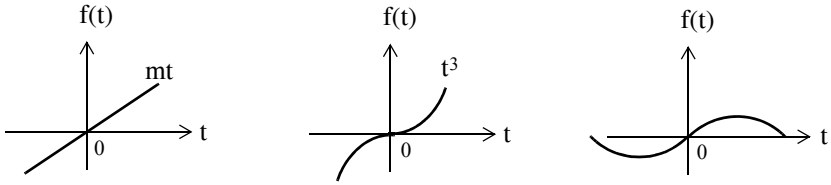


Figure 6.5. Examples of odd functions

We observe that for odd functions, $f(0) = 0$. However, the reverse is not always true; that is, if $f(0) = 0$, we should not conclude that $f(t)$ is an odd function. An example of this is the function $f(t) = t^2$ in Figure 6.4.

The product of *two even* or *two odd* functions is an even function, and the product of an even function times an odd function, is an odd function.

Henceforth, we will denote an even function with the subscript e, and an odd function with the subscript o. Thus, $f_e(t)$ and $f_o(t)$ will be used to represent even and odd functions of time respectively.

For an even function $f_e(t)$,

$$\int_{-T}^{T} f_e(t)dt = 2\int_{0}^{T} f_e(t)dt \tag{6.14}$$

and for an odd function $f_o(t)$,

$$\int_{-T}^{T} f_o(t)dt = 0 \tag{6.15}$$

A function $f(t)$ that is neither even nor odd can be expressed as

$$\boxed{f_e(t) = \frac{1}{2}[f(t) + f(-t)]} \tag{6.16}$$

or as

$$\boxed{f_o(t) = \frac{1}{2}[f(t) - f(-t)]} \tag{6.17}$$

Addition of (6.16) with (6.17) yields

$$\boxed{f(t) = f_e(t) + f_o(t)} \tag{6.18}$$

that is, any function of time can be expressed as the sum of an even and an odd function.

Example 6.3

Determine whether the delta function is an even or an odd function of time.

Solution:

Let $f(t)$ be an arbitrary function of time that is continuous at $t = t_0$. Then, by the sifting property of the delta function

$$\int_{-\infty}^{\infty} f(t)\delta(t - t_0)dt = f(t_0)$$

and for $t_0 = 0$,

$$\int_{-\infty}^{\infty} f(t)\delta(t)dt = f(0)$$

Also,

$$\int_{-\infty}^{\infty} f_e(t)\delta(t)dt = f_e(0)$$

and

$$\int_{-\infty}^{\infty} f_o(t)\delta(t)dt = f_o(0)$$

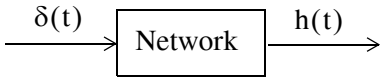
As stated earlier, an odd function $f_o(t)$ evaluated at $t = 0$ is zero, that is, $f_o(0) = 0$. Therefore, from the last relation above,

$$\int_{-\infty}^{\infty} f_o(t)\delta(t)dt = f_o(0) = 0 \tag{6.19}$$

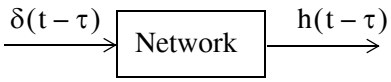
and this indicates that the product $f_o(t)\delta(t)$ is an odd function of t. Then, since $f_o(t)$ is odd, it follows that $\delta(t)$ must be an even function of t for (6.19) to hold.

6.3 Convolution

Consider a network whose input is $\delta(t)$, and its output is the impulse response $h(t)$. We can represent the input–output relationship as the block diagram shown below.

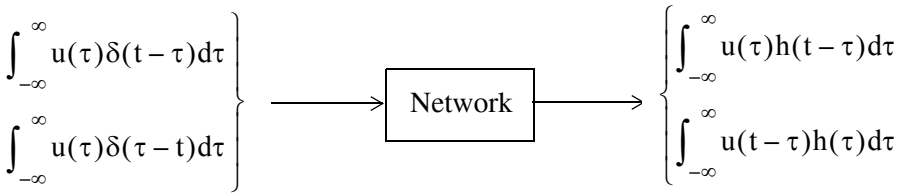


In general,

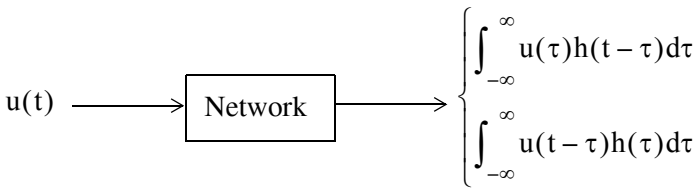


Next, we let $u(t)$ be any input whose value at $t = \tau$ is $u(\tau)$. Then,

Multiplying both sides by the constant $d\tau$, integrating from $-\infty$ to $+\infty$, and making use of the fact that the delta function is even, i.e., $\delta(t-\tau) = \delta(\tau - t)$, we obtain



Using the sifting property of the delta function, we find that the second integral on the left side reduces to $u(t)$ and thus



The integral

$$\boxed{\int_{-\infty}^{\infty} u(\tau)h(t-\tau)d\tau} \quad \text{or} \quad \boxed{\int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau} \tag{6.20}$$

is known as the *convolution integral*; it states that if we know the impulse response of a network, we can compute the response to any input $u(t)$ using either of the integrals of (6.20).

The convolution integral is usually represented as $u(t)*h(t)$ or $h(t)*u(t)$, where the asterisk (*) denotes convolution.

In Section 6.1, we found that the impulse response for a single input is $h(t) = e^{At}b$. Therefore, if we know $h(t)$, we can use the convolution integral to compute the response $y(t)$ of any input $u(t)$ using the relation

$$y(t) = \int_{-\infty}^{\infty} e^{A(t-\tau)}bu(\tau)d\tau = e^{At}\int_{-\infty}^{\infty} e^{-A\tau}bu(\tau)d\tau \tag{6.21}$$

6.4 Graphical Evaluation of the Convolution Integral

The convolution integral is more conveniently evaluated by the graphical evaluation. The procedure is best illustrated with the following examples.

Example 6.4

The signals $h(t)$ and $u(t)$ are as shown in Figure 6.6. Compute $h(t)*u(t)$ using the graphical evaluation.

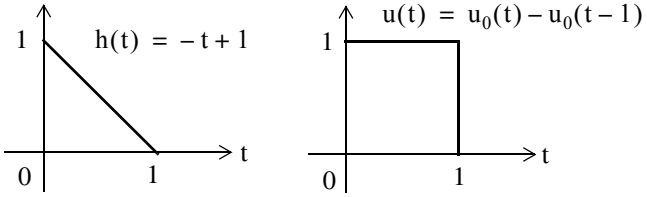


Figure 6.6. Signals for Example 6.4

Solution:

The convolution integral states that

$$h(t)*u(t) = \int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau \tag{6.22}$$

where τ is a dummy variable, that is, $u(\tau)$ and $h(\tau)$, are considered to be the same as $u(t)$ and $h(t)$. We form $u(t-\tau)$ by first constructing the image of $u(\tau)$; this is shown as $u(-\tau)$ in Figure 6.7.

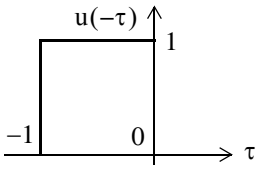


Figure 6.7. Construction of $u(-\tau)$ for Example 6.4

Next, we form $u(t-\tau)$ by shifting $u(-\tau)$ to the right by some value t as shown in Figure 6.8.

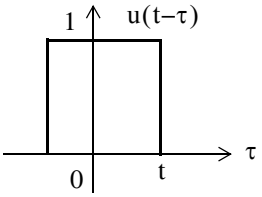


Figure 6.8. Formation of $u(t-\tau)$ for Example 6.4

Now, evaluation of the convolution integral

$$h(t)*u(t) = \int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau$$

entails multiplication of $u(t-\tau)$ by $h(\tau)$ for each value of t, and computation of the area from $-\infty$ to $+\infty$. Figure 6.9 shows the product $u(t-\tau)h(\tau)$ as point A moves to the right.

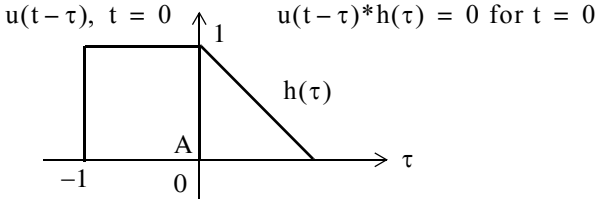


Figure 6.9. Formation of the product $u(t-\tau)*h(\tau)$ for Example 6.4

We observe that $u(t-\tau)|_{t=0} = u(-\tau)$. Shifting $u(t-\tau)$ to the right so that $t>0$, we obtain the sketch of Figure 6.10 where the integral of the product is denoted by the shaded area, and it increases as point A moves further to the right.

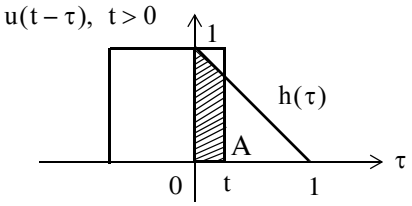


Figure 6.10. Shift of $u(t-\tau)$ for Example 6.4

The maximum area is obtained when point A reaches $t = 1$ as shown in Figure 6.11.

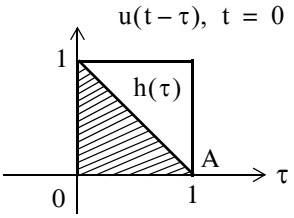


Figure 6.11. Signals for Example 6.4 when $t = 1$

Using the convolution integral, we find that the area as a function of time t is

$$\int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau = \int_{0}^{t} u(t-\tau)h(\tau)d\tau = \int_{0}^{t} (1)(-\tau+1)d\tau = \tau - \frac{\tau^2}{2}\bigg|_{0}^{t} = t - \frac{t^2}{2} \tag{6.23}$$

Figure 6.12 shows how $u(\tau)*h(\tau)$ increases during the interval $0 < t < 1$. This is not an exponential increase; it is the function $t - t^2/2$ in (6.23), and each point on the curve of Figure 6.12 represents the area under the convolution integral.

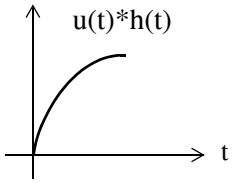


Figure 6.12. Curve for the convolution of $u(\tau)*h(\tau)$ for $0 < t < 1$ in Example 6.4

Evaluating (6.23) at $t = 1$, we obtain

$$t - \frac{t^2}{2}\bigg|_{t=1} = \frac{1}{2} \tag{6.24}$$

The plot for the interval $0 \le t \le 1$ is shown in Figure 6.13.

As we continue shifting $u(t-\tau)$ to the right, the area starts decreasing, and it becomes zero at $t = 2$, as shown in Figure 6.14.

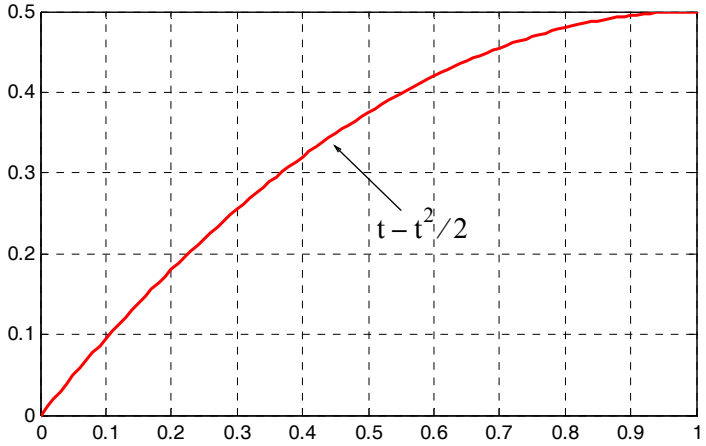


Figure 6.13. Convolution of $u(\tau)*h(\tau)$ at $t = 1$ for Example 6.4

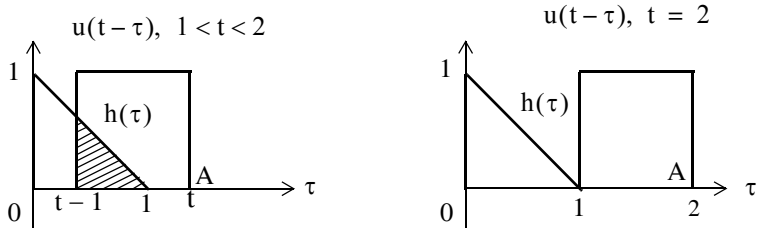


Figure 6.14. Convolution for interval $1 < t < 2$ of Example 6.4

Using the convolution integral, we find that the area for the interval $1 < t < 2$ is

$$\int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau = \int_{t-1}^{1} u(t-\tau)h(\tau)d\tau = \int_{t-1}^{1} (1)(-\tau+1)d\tau = \tau - \frac{\tau^2}{2}\Big|_{t-1}^{1}$$

$$= 1 - \frac{1}{2} - (t-1) + \frac{t^2-2t+1}{2} = \frac{t^2}{2} - 2t + 2 \qquad (6.25)$$

Thus, for $1 < t < 2$, the area decreases in accordance with $t^2/2 - 2t + 2$.

Evaluating (6.25) at $t = 2$, we find that $u(\tau)*h(\tau) = 0$. For $t > 2$, the product $u(t-\tau)h(\tau)$ is zero since there is no overlap between these two signals. The convolution of these signals for $0 \le t \le 2$, is shown in Figure 6.15.

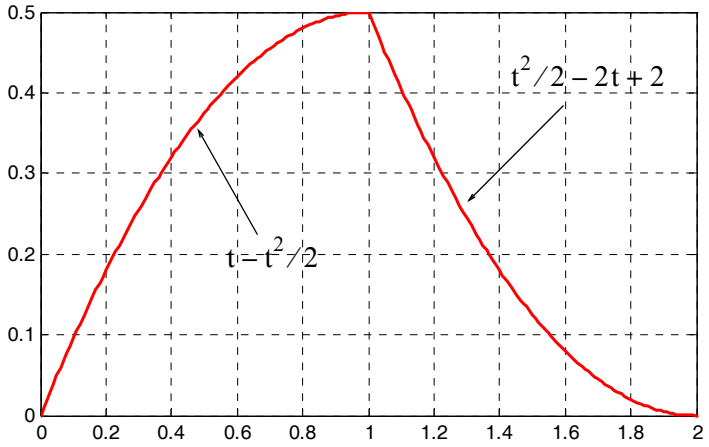


Figure 6.15. Convolution for $0 \le \tau \le 2$ of the signals of Example 6.4

The plot of Figure 6.15 was obtained with the MATLAB script below.

```
t1=0:0.01:1; x=t1–t1.^2./2; axis([0  1  0  0.5]);...
t2=1:0.01:2; y=t2.^2./2–2.*t2+2; axis([1  2  0  0.5]); plot(t1,x,t2,y); grid
```

Example 6.5

The signals h(t) and u(t) are as shown in Figure 6.16. Compute h(t)*u(t) using the graphical evaluation method.

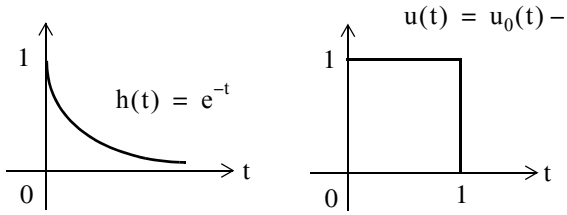


Figure 6.16. Signals for Example 6.5

Solution:

Following the same procedure as in the previous example, we form $u(t-\tau)$ by first constructing the image of $u(\tau)$. This is shown as $u(-\tau)$ in Figure 6.17.

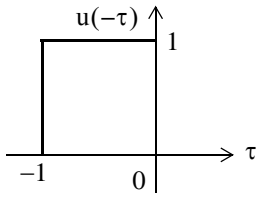


Figure 6.17. Construction of $u(-\tau)$ for Example 6.5

Next, we form $u(t-\tau)$ by shifting $u(-\tau)$ to the right by some value t as shown in Figure 6.18.

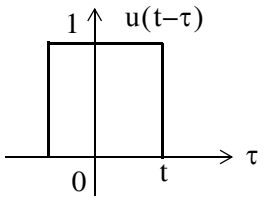


Figure 6.18. Formation of $u(t-\tau)$ for Example 6.5

As in the previous example, evaluation of the convolution integral

$$h(t)*u(t) = \int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau$$

entails multiplication of $u(t-\tau)$ by $h(\tau)$ for each value ot t, and computation of the area from $-\infty$ to $+\infty$. Figure 6.19 shows the product $u(t-\tau)h(\tau)$ as point A moves to the right.

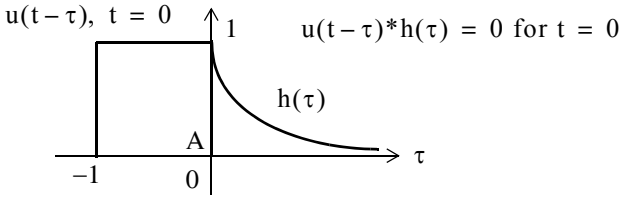


Figure 6.19. Formation of the product $u(t-\tau)*h(\tau)$ for Example 6.5

We observe that $u(t-\tau)|_{t=0} = u(-\tau)$. Shifting $u(t-\tau)$ to the right so that $t>0$, we obtain the sketch of Figure 6.20 where the integral of the product is denoted by the shaded area, and it increases as point A moves further to the right.

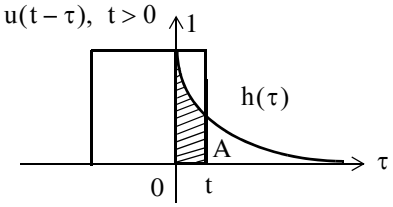


Figure 6.20. Shift of $u(t-\tau)$ for Example 6.5

The maximum area is obtained when point A reaches $t = 1$ as shown in Figure 6.21.

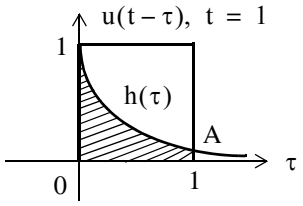


Figure 6.21. Convolution of $u(\tau)*h(\tau)$ at $t = 1$ for Example 6.5

Its value for $0 < t < 1$ is

$$\int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau = \int_{0}^{t} u(t-\tau)h(\tau)d\tau = \int_{0}^{t} (1)(e^{-\tau})d\tau = -e^{-\tau}\Big|_{0}^{t} = e^{-\tau}\Big|_{t}^{0} = 1-e^{-t} \tag{6.26}$$

Evaluating (6.26) at $t = 1$, we obtain

$$1-e^{-t}\Big|_{t=1} = 1-e^{-1} = 0.632 \tag{6.27}$$

The plot for the interval $0 \le t \le 1$ is shown in Figure 6.22.

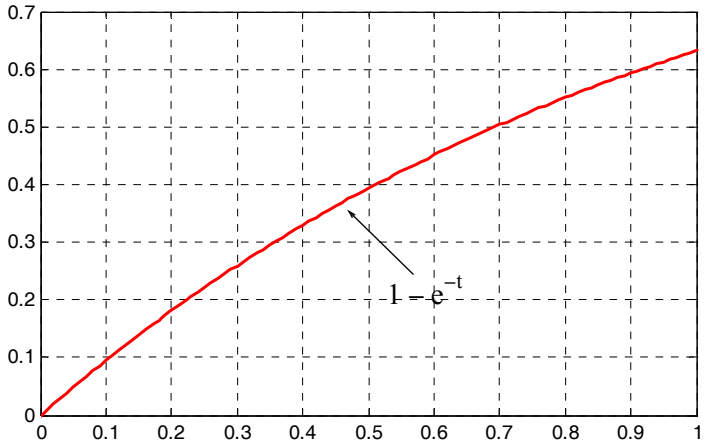


Figure 6.22. Convolution of $u(\tau)*h(\tau)$ for $0 \le t \le 1$ in Example 6.5

As we continue shifting $u(t-\tau)$ to the right, the area starts decreasing. As shown in Figure 6.23, it approaches zero as t becomes large but never reaches the value of zero.

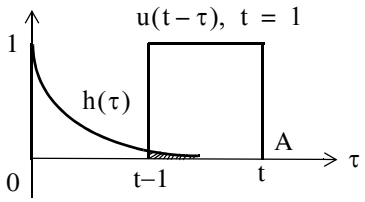


Figure 6.23. Convolution for interval $1 < t < 2$ of Example 6.5

Therefore, for the time interval $t > 1$, we have

$$\int_{t-1}^{t} u(t-\tau)h(\tau)d\tau = \int_{t-1}^{t} (1)(e^{-\tau})d\tau = -e^{-\tau}\Big|_{t-1}^{t} = e^{-\tau}\Big|_{t}^{t-1} = e^{-(t-1)} - e^{-t} = e^{-t}(e-1) \tag{6.28}$$

$$= 1.732e^{-t}$$

Evaluating (6.28) at $t = 2$, we find that $u(\tau)*h(\tau) = 0.233$.

For $t > 2$, the product $u(t-\tau)h(\tau)$ approaches zero as $t \to \infty$. The convolution of these signals for $0 \le t \le 2$, is shown in Figure 6.24.

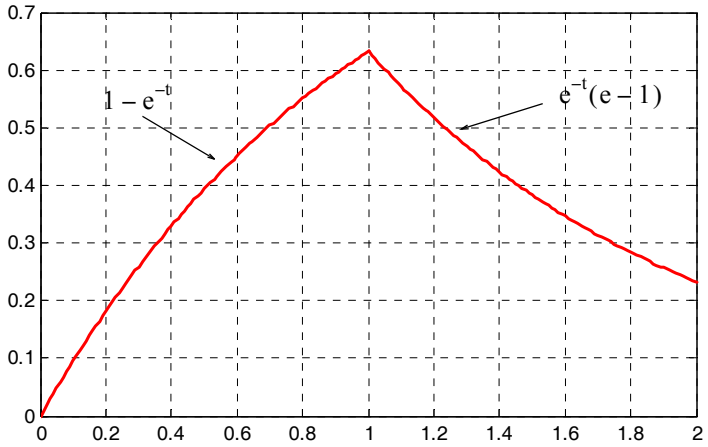


Figure 6.24. Convolution for $0 \le t \le 2$ of the signals of Example 6.5

The plot of Figure 6.24 was obtained with the MATLAB script below.

```
t1=0:0.01:1; x=1–exp(–t1); axis([0 1 0 0.8]);...
t2=1:0.01:2; y=1.718.*exp(–t2); axis([1 2 0 0.8]); plot(t1,x,t2,y); grid
```

Example 6.6

Perform the convolution $v_1(t)*v_2(t)$ where $v_1(t)$ and $v_2(t)$ are as shown in Figure 6.25.

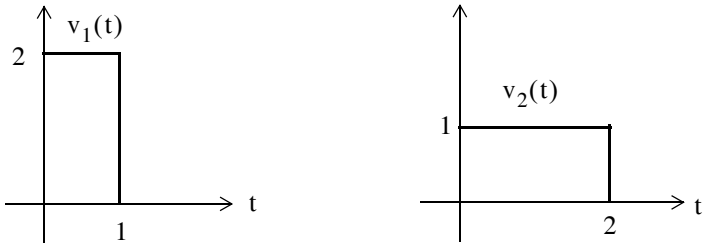


Figure 6.25. Signals for Example 6.6

Solution:

We will use the convolution integral

$$v_1(t)*v_2(t) = \int_{-\infty}^{\infty} v_1(\tau)v_2(t-\tau)d\tau \tag{6.29}$$

The computation steps are as in the two previous examples, and are evident from the sketches of Figures 6.26 through 6.29.

Figure 6.26 shows the formation of $v_2(-\tau)$.

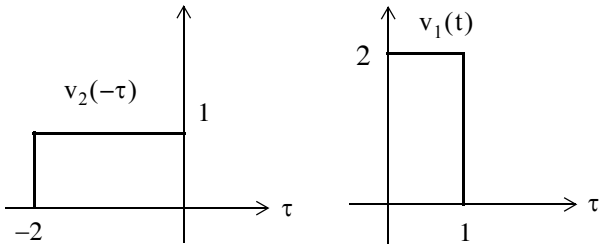


Figure 6.26. Formation of $v_2(-\tau)$ for Example 6.6

Figure 6.27 shows the formation of $v_2(t-\tau)$ and convolution with $v_1(t)$ for $0 < t < 1$.

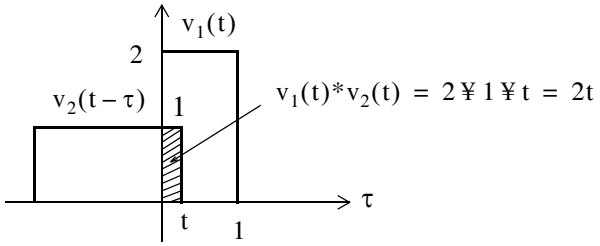


Figure 6.27. Formation of $v_2(t-\tau)$ and convolution with $v_1(t)$

For $0 < t < 1$,

$$v_1(t)*v_2(t) = \int_0^t (1)(2)d\tau = 2\tau\Big|_0^t = 2t \tag{6.30}$$

Figure 6.28 shows the convolution of $v_2(t-\tau)$ with $v_1(t)$ for $1<t<2$.

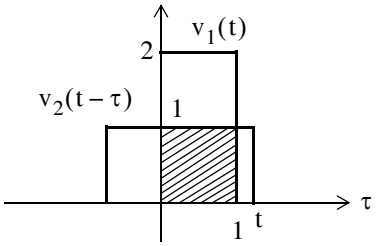


Figure 6.28. Convolution of $v_2(t-\tau)$ with $v_1(t)$ for $1<t<2$

For $1<t<2$,

$$v_1(t)*v_2(t) = \int_0^1 (1)(2)d\tau = 2\tau\Big|_0^1 = 2 \tag{6.31}$$

Figure 6.29 shows the convolution of $v_2(t-\tau)$ with $v_1(t)$ for $2<t<3$.

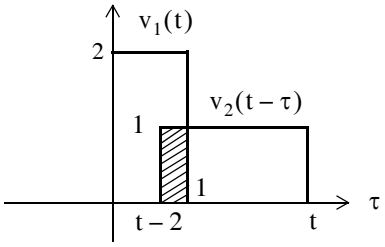


Figure 6.29. Convolution of $v_2(t-\tau)$ with $v_1(t)$ for $2<t<3$

For $2<t<3$

$$v_1(t)*v_2(t) = \int_{t-2}^1 (1)(2)d\tau = 2\tau\Big|_{t-2}^1 = -2t+6 \tag{6.32}$$

From (6.30), (6.31), and (6.32), we obtain the waveform of Figure 6.30 that represents the convolution of the signals $v_1(t)$ and $v_2(t-\tau)$.

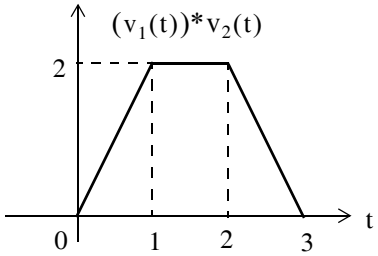


Figure 6.30. Convolution of $v_1(t)$ with $v_2(t)$ for $0<t<3$

In summary, the procedure for the graphical evaluation of the convolution integral, is as follows:

1. We substitute $u(t)$ and $h(t)$ with $u(\tau)$ and $h(\tau)$ respectively.
2. We fold (form the mirror image of) $u(\tau)$ or $h(\tau)$ about the vertical axis to obtain $u(-\tau)$ or $h(-\tau)$.
3. We slide $u(-\tau)$ or $h(-\tau)$ to the right a distance t to obtain $u(t-\tau)$ or $h(t-\tau)$.
4. We multiply the two functions to obtain the product $u(t-\tau)\,h(\tau)$, or $u(\tau)\,h(t-\tau)$.
5. We integrate this product by varying t from $-\infty$ to $+\infty$.

6.5 Circuit Analysis with the Convolution Integral

We can use the convolution integral in circuit analysis as illustrated by the following example.

Example 6.7

For the circuit of Figure 6.31, use the convolution integral to find the capacitor voltage when the input is the unit step function $u_0(t)$, and $v_C(0^-) = 0$.

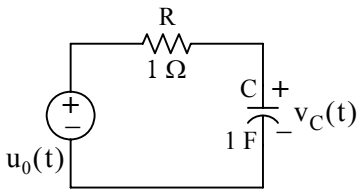


Figure 6.31. Circuit for Example 6.7

Solution:

Before we apply the convolution integral, we must know the impulse response $h(t)$ of this circuit. The circuit of Figure 6.31 was analyzed in Example 6.1, Page 6–2, where we found that

$$h(t) = \frac{1}{RC}\,e^{-t/RC}u_0(t) \tag{6.33}$$

With the given values, (6.33) reduces to

$$h(t) = e^{-t}u_0(t) \tag{6.34}$$

Next, we use the graphical evaluation of the convolution integral as shown in Figures 6.32 through 6.34.

The formation of $u_0(-\tau)$ is shown in Figure 6.32.

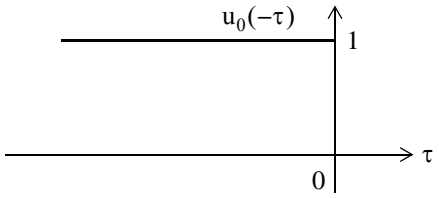


Figure 6.32. Formation of $u_0(-\tau)$ for Example 6.7

Figure 6.33 shows the formation of $u_0(t-\tau)$.

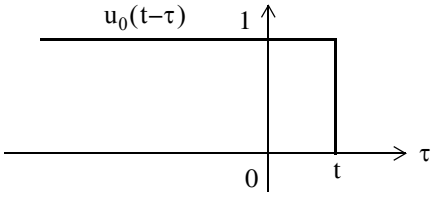


Figure 6.33. Formation of $u_0(t-\tau)$ for Example 6.7

Figure 6.34 shows the convolution $(u_0(t))*h(t)$.

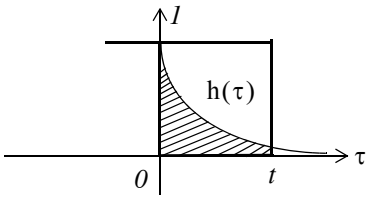


Figure 6.34. Convolution of $u_0(t)*h(t)$ for Example 6.7

Therefore, for the interval $0 < t < \infty$, we obtain

$$u_0(t)*h(t) = \int_{-\infty}^{\infty} u_0(t-\tau)h(\tau)d\tau = \int_0^t (1)e^{-t}d\tau = -e^{-t}\Big|_0^t = e^{-t}\Big|_t^0 = (1-e^{-t})u_0(t) \qquad (6.35)$$

and the convolution $u_0(t)*h(t)$ is shown in Figure 6.35.

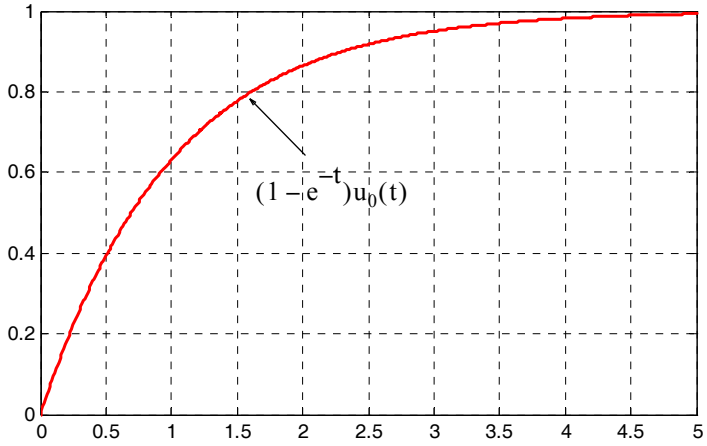


Figure 6.35. Convolution of $u_0(t)*h(t)$ for Example 6.7

6.6 Summary

- The impulse response is the output (voltage or current) of a network when the input is the delta function.
- The determination of the impulse response assumes zero initial conditions.
- A function $f(t)$ is an even function of time if the following relation holds.

$$f(-t) = f(t)$$

- A function $f(t)$ is an odd function of time if the following relation holds.

$$-f(-t) = f(t)$$

- The product of two even or two odd functions is an even function, and the product of an even function times an odd function, is an odd function.
- A function $f(t)$ that is neither even nor odd, can be expressed as

$$f_e(t) = \frac{1}{2}[f(t) + f(-t)]$$

or as

$$f_o(t) = \frac{1}{2}[f(t) - f(-t)]$$

where $f_e(t)$ denotes an even function and $f_o(t)$ denotes an odd function.

- Any function of time can be expressed as the sum of an even and an odd function, that is,

$$f(t) = f_e(t) + f_o(t)$$

- The delta function is an even function of time.
- The integral

$$\int_{-\infty}^{\infty} u(\tau)h(t-\tau)d\tau$$

or

$$\int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau$$

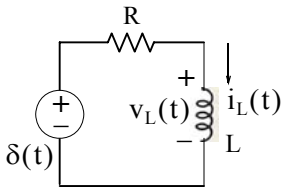
is known as the convolution integral.

- If we know the impulse response of a network, we can compute the response to any input $u(t)$ with the use of the convolution integral.

- The convolution integral is usually denoted as `u(t)*h(t)` or `h(t)*u(t)`, where the asterisk (*) denotes convolution.
- The convolution integral is more conveniently evaluated by the graphical evaluation method.

6.7 Exercises

1. Compute the impulse response $h(t) = i_L(t)$ in terms of R and L for the circuit below. Then, compute the voltage $v_L(t)$ across the inductor.



2. Repeat Example 6.4, Page 6–8, by forming $h(t-\tau)$ instead of $u(t-\tau)$, that is, use the convolution integral

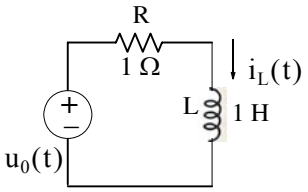
$$\int_{-\infty}^{\infty} u(\tau)h(t-\tau)d\tau$$

3. Repeat Example 6.5, Page 6–12, by forming $h(t-\tau)$ instead of $u(t-\tau)$.

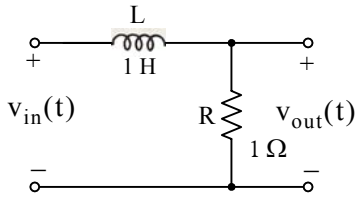
4. Compute $v_1(t)*v_2(t)$ given that

$$v_1(t) = \begin{cases} 4t & t \geq 0 \\ 0 & t < 0 \end{cases}$$

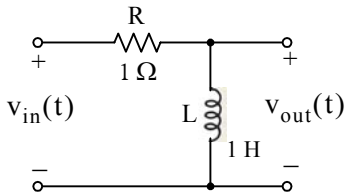
5. For the series RL circuit shown below, the response is the current $i_L(t)$. Use the convolution integral to find the response when the input is the unit step $u_0(t)$.



6. Compute $v_{out}(t)$ for the network shown below using the convolution integral, given that $v_{in}(t) = u_0(t) - u_0(t-1)$.



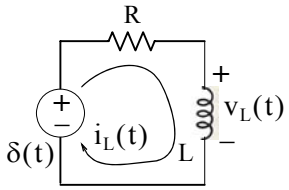
7. Compute $v_{out}(t)$ for the network shown below given that $v_{in}(t) = u_0(t) - u_0(t-1)$. Using MATLAB, plot $v_{out}(t)$ for the time interval $0 < t < 5$.



Hint: Use the result of Exercise 6.

6.8 Solutions to End–of–Chapter Exercises

1.



$$Ri + L\frac{di}{dt} = \delta(t)$$

Letting state variable $x = i$, the above relation is written as

$$\dot{x} = (-R/L)x + (1/L)\delta(t)$$

and this has the form $\dot{x} = Ax + bu$ where $A = -R/L$, $b = 1/L$, and $u = \delta(t)$. Its solution is

$$x(t) = e^{A(t-t_0)}x_0 + e^{At}\int_0^t e^{-A\tau}bu(\tau)d\tau$$

and from (6.5), Page 6–1,

$$h(t) = i(t) = e^{At}bu_0(t) = e^{-(R/L)t}\cdot 1/L \cdot u_0(t) = (1/L)e^{-(R/L)t}u_0(t)$$

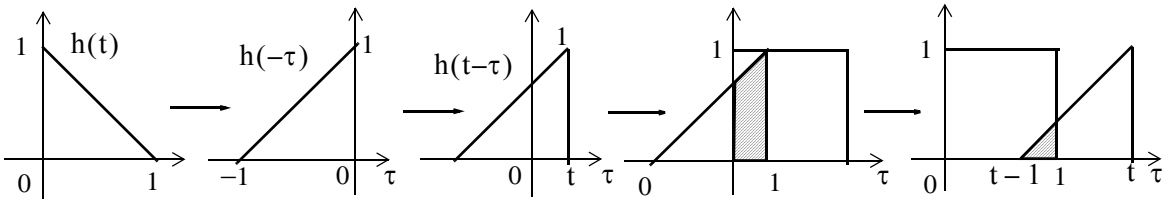
The voltage v_L across the inductor is found from

$$v_L = L\frac{d}{dt}i(t) = L\frac{d}{dt}h(t) = L\frac{d}{dt}\left(\frac{1}{L}e^{-(R/L)t}u_0(t)\right) = (-R/L)e^{-(R/L)t}u_0(t) + e^{-(R/L)t}\delta(t)$$

and using the sampling property of the delta function, the above relation reduces to

$$v_L = (-R/L)e^{-(R/L)t}u_0(t) + \delta(t)$$

2.



From the plots above we observe that the area reaches the maximum value of $1/2$ at $t = 1$, and then decreases to zero at $t = 2$. Alternately, using the convolution integral we obtain

$$u(t)*h(t) = \int_{-\infty}^{\infty} u(\tau)h(t-\tau)d\tau$$

where $h(t) = -t+1$, $h(\tau) = -\tau+1$, $h(-\tau) = \tau+1$, and $h(t-\tau) = -(t-\tau)+1 = 1-t+\tau$. Then, for $0<t<1$,

$$\text{Area}_1 = \int_0^t [(1-t)+\tau]d\tau = \frac{\tau^2}{2} + (1-t)\tau\Big|_0^t = \frac{t^2}{2} + (1-t)t = t - \frac{t^2}{2}$$

and we observe that at $t = 1$, $\text{Area}_1 = 1/2$ square units

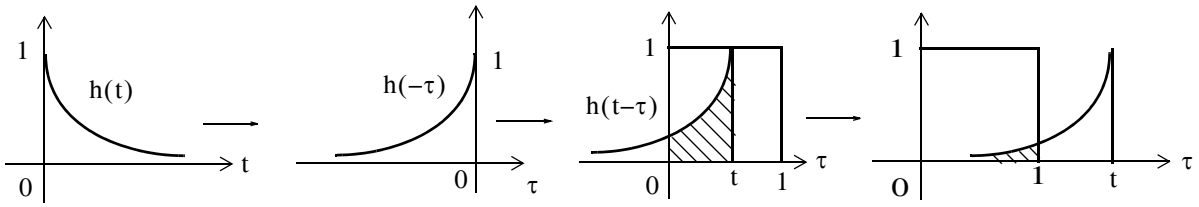
Next, for $1<t<2$,

$$\text{Area}_2 = \int_{t-1}^{1} [(1-t)+\tau]d\tau = \frac{\tau^2}{2} + (1-t)\tau\Big|_{t-1}^{1}$$

$$= \frac{1}{2} + 1 - t - \frac{(t-1)^2}{2} - (1-t)\cdot(t-1) = \frac{t^2}{2} - 2t + 2$$

and we observe that at $t = 2$, $\text{Area}_2 = 0$

3.



From the plots above we observe that the area reaches its maximum value at $t = 1$, and then decreases exponentially to zero as $t \to \infty$. Alternately, using the convolution integral we obtain

$$u(t)*h(t) = \int_{-\infty}^{\infty} u(\tau)h(t-\tau)d\tau$$

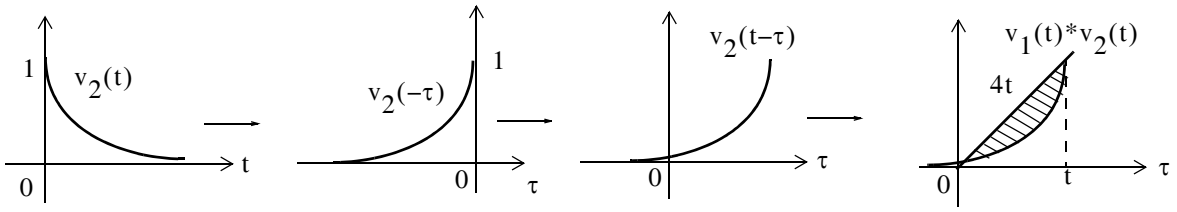
where $h(t) = e^{-t}$, $h(\tau) = e^{-\tau}$, $h(-\tau) = e^{\tau}$, and $h(t-\tau) = e^{-(t-\tau)}$. Then, for $0<t<1$

$$\text{Area}_1 = \int_0^t (1\cdot e^{-(t-\tau)})d\tau = e^{-t}\int_0^t e^{\tau}d\tau = e^{-t}(e^t - e^0) = 1 - e^{-t}$$

For $t>1$,

$$\text{Area}_2 = \int_0^1 (1\cdot e^{-(t-\tau)})d\tau = e^{-t}\int_0^1 e^{\tau}d\tau = e^{-t}e^{\tau}\Big|_0^1 = e^{-t}(e^1 - e^0) = e^{-t}(e-1) = 1.732e^{-t}$$

4.



$$v_1(t)*v_2(t) = \int_0^t v_1(\tau)v_2(t-\tau)d\tau = \int_0^t 4\tau e^{-2(t-\tau)}d\tau = 4e^{-2t}\int_0^t \tau e^{2\tau}d\tau$$

From tables of integrals,

$$\int xe^{ax}dx = \frac{e^{ax}}{a^2}(ax-1)$$

and thus

$$v_1(t)*v_2(t) = (4e^{-2t})\frac{e^{2\tau}(2\tau-1)}{4}\Bigg|_0^t = e^{-2t}[e^{2t}(2t-1)-e^0(-1)]$$

$$= e^0(2t-1+e^{-2t}) = 2t+e^{-2t}-1$$

Check:

$$v_1(t)*v_2(t) \Leftrightarrow V_1(s)\cdot V_2(s),\ V_1(s) = 4/s^2,\ V_2(s) = 1/(s+2)$$

$$V_1(s)\cdot V_2(s) = \frac{4}{s^2(s+2)} = \frac{4}{s^3+2s^2}$$

```
syms s t; ilaplace(4/(s^3+2*s^2))

ans =
2*t-1+exp(-2*t)
```

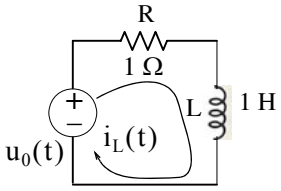
5.

To use the convolution integral, we must first find the impulse response. It was found in Exercise 1 as

$$h(t) = i(t) = (1/L)e^{-(R/L)t}u_0(t)$$

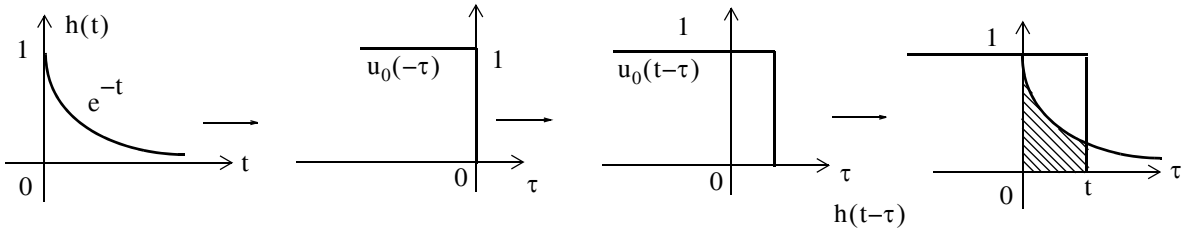
and with the given values,

$$h(t) = e^{-t}u_0(t)$$

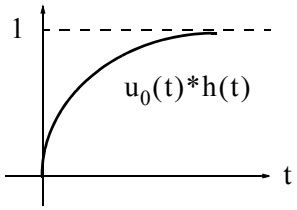


When the input is the unit step $u_0(t)$,

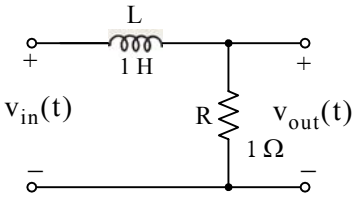
$$i(t)\big|_{v_{in}=u_0(t)} = \int_{-\infty}^{\infty} u_0(t-\tau)h(\tau)d\tau$$



$$i(t)\big|_{v_{in}=u_0(t)} = \int_0^t (1)\cdot e^{-\tau}d\tau = -e^{-\tau}\big|_0^t = e^{-\tau}\big|_t^0 = (1-e^{-t})u_0(t)$$



6.



We will first compute the impulse response, that is, the output when the input is the delta function, i.e., $v_{in}(t) = \delta(t)$. Then, by KVL

$$L\frac{di_L}{dt} + Ri_L = \delta(t)$$

and with $i_L = x$

$$1\cdot\dot{x} + 1\cdot x = \delta(t)$$

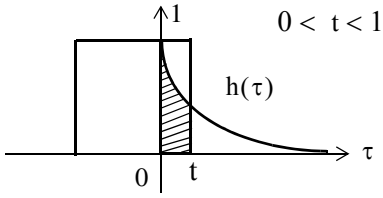
or

$$\dot{x} = -x + \delta(t)$$

By comparison with $\dot{x} = Ax + bu$, we observe that $A = -1$ and $b = 1$.
From (6.5)

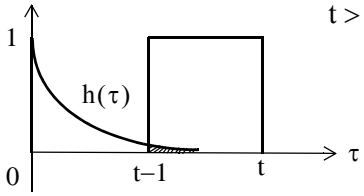
$$h(t) = e^{At}bu_0(t) = e^{-t} \cdot 1 = e^{-t}$$

Now, we compute $v_{out}(t)$ when $v_{in}(t) = u_0(t) - u_0(t-1)$ by convolving the impulse response $h(t)$ with this input $v_{in}(t)$, that is, $v_{out}(t) = v_{in}(t)*h(t)$. The remaining steps are as in Example 6.5 and are shown below.



$0 < t < 1$

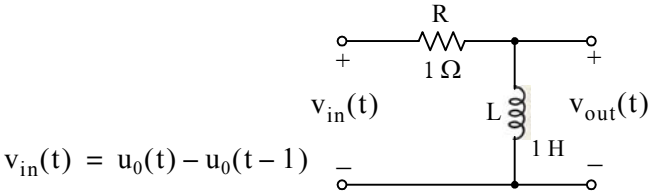
$$\int_0^t (1)(e^{-\tau})d\tau = -e^{-\tau}\Big|_0^t = e^{-\tau}\Big|_t^0 = 1 - e^{-t}$$



$t > 1$

$$\int_{t-1}^t (1)(e^{-\tau})d\tau = -e^{-\tau}\Big|_{t-1}^t = e^{-\tau}\Big|_t^{t-1} = e^{-t}(e-1)$$

7.



$$v_{in}(t) = u_0(t) - u_0(t-1)$$

$$v_{out}(t) = v_L = v_{in} - v_R$$

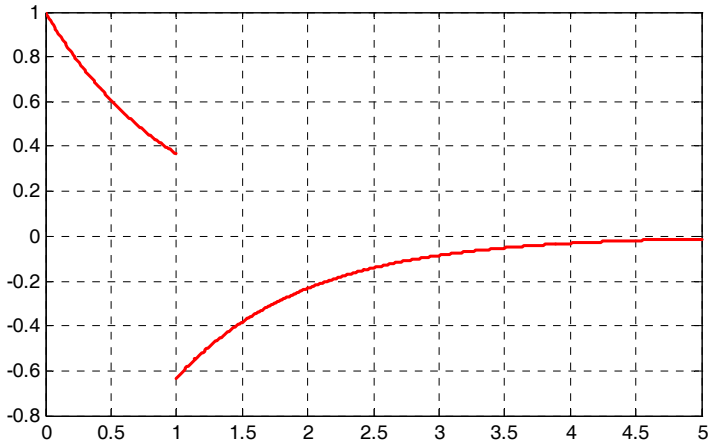
From Exercise 6,

$$v_R = \begin{cases} 1 - e^{-t} & 0 < t < 1 \\ e^{-t}(e-1) & t > 1 \end{cases}$$

Then, for this circuit,

$$v_{out} = v_L = \begin{cases} (1 - (1 - e^{-t}) = e^{-t}) & 0 < t < 1 \\ 0 - e^{-t}(e-1) = (1-e)e^{-t} = -1.732e^{-t} & t > 1 \end{cases}$$

The plot for the time interval $0 < t < 5$ is shown below.



The plot above was obtained with the MATLAB script below.

```
t1=0:0.01:1; x=exp(–t1); axis([0 1 0 1]);...
t2=1:0.01:5; y=–1.718.*exp(–t2); axis([1 5 0 1]); plot(t1,x,t2,y); grid
```

Chapter 7

Fourier Series

This chapter is an introduction to Fourier series. We begin with the definition of sinusoids that are harmonically related and the procedure for determining the coefficients of the trigonometric form of the series. Then, we discuss the different types of symmetry and how they can be used to predict the terms that may be present. Several examples are presented to illustrate the approach. The alternate trigonometric and the exponential forms are also presented.

7.1 Wave Analysis

The French mathematician Fourier found that any *periodic* waveform, that is, a waveform that repeats itself after some time, can be expressed as a series of harmonically related sinusoids, i.e., sinusoids whose frequencies are multiples of a *fundamental* frequency (or first harmonic). For example, a series of sinusoids with frequencies 1 MHz, 2 MHz, 3 MHz, and so on, contains the fundamental frequency of 1 MHz, a second harmonic of 2 MHz, a third harmonic of 3 MHz, and so on. In general, any periodic waveform $f(t)$ can be expressed as

$$f(t) = \frac{1}{2}a_0 + a_1\cos\omega t + a_2\cos 2\omega t + a_3\cos 3\omega t + a_4\cos 4\omega t + \ldots + b_1\sin\omega t + b_2\sin 2\omega t + b_3\sin 3\omega t + b_4\sin 4\omega t + \ldots \tag{7.1}$$

or

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty}(a_n\cos n\omega t + b_n\sin n\omega t) \tag{7.2}$$

where the first term $a_0/2$ is a constant, and represents the DC (average) component of $f(t)$. Thus, if $f(t)$ represents some voltage $v(t)$, or current $i(t)$, the term $a_0/2$ is the average value of $v(t)$ or $i(t)$.

The terms with the coefficients a_1 and b_1 together, represent the fundamental frequency component ω*. Likewise, the terms with the coefficients a_2 and b_2 together, represent the second harmonic component 2ω, and so on.

Since any periodic waveform $f(t))$ can be expressed as a Fourier series, it follows that the sum of

* *We recall that* $k_1\cos\omega t + k_2\sin\omega t = k\cos(\omega t + \theta)$ *where* θ *is a constant.*

the DC, the fundamental, the second harmonic, and so on, must produce the waveform f(t). Generally, the sum of two or more sinusoids of different frequencies produce a waveform that is not a sinusoid as shown in Figure 7.1.

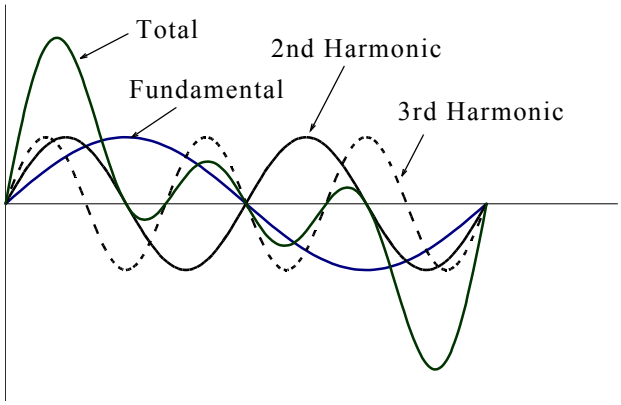


Figure 7.1. Summation of a fundamental, second and third harmonic

7.2 Evaluation of the Coefficients

Evaluations of a_i and b_i coefficients of (7.1) is not a difficult task because the sine and cosine are *orthogonal functions*, that is, the product of the sine and cosine functions under the integral evaluated from 0 to 2π is zero. This will be shown shortly.

Let us consider the functions $\sin mt$ and $\cos mt$ where m and n are any integers. Then,

$$\int_0^{2\pi} \sin mt\,dt = 0 \tag{7.3}$$

$$\int_0^{2\pi} \cos mt\,dt = 0 \tag{7.4}$$

$$\int_0^{2\pi} (\sin mt)(\cos nt)\,dt = 0 \tag{7.5}$$

The integrals of (7.3) and (7.4) are zero since the net area over the 0 to 2π area is zero. The integral of (7.5) is also is zero since

$$\sin x \cos y = \frac{1}{2}[\sin(x+y) + \sin(x-y)]$$

This is also obvious from the plot of Figure 7.2, where we observe that the net shaded area above and below the time axis is zero.

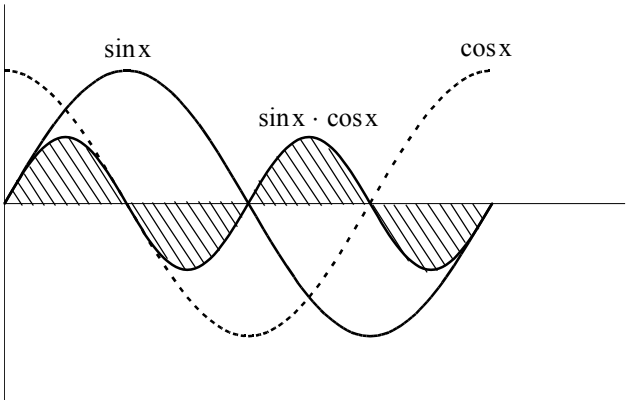


Figure 7.2. *Graphical proof of* $\int_0^{2\pi}(\sin mt)(\cos nt)dt = 0$

Moreover, if m and n are different integers, then,

$$\int_0^{2\pi}(\sin mt)(\sin nt)dt = 0 \tag{7.6}$$

since

$$(\sin x)(\sin y) = \frac{1}{2}[\cos(x-y) - \cos(x-y)]$$

The integral of (7.6) can also be confirmed graphically as shown in Figure 7.3, where $m = 2$ and $n = 3$. We observe that the net shaded area above and below the time axis is zero.

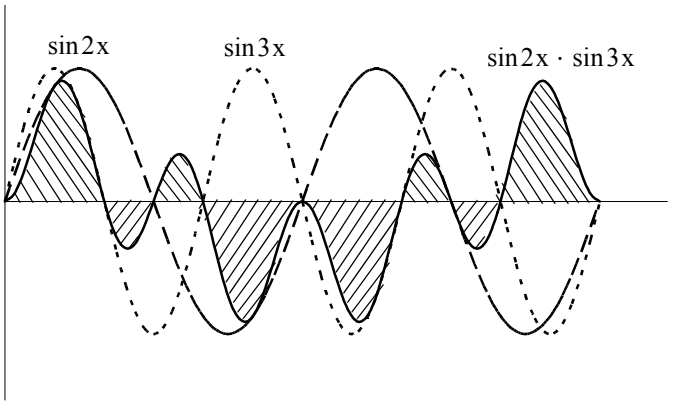


Figure 7.3. Graphical proof of $\int_0^{2\pi}(\sin mt)(\sin nt)dt = 0$ for $m = 2$ and $n = 3$

Also, if m and n are different integers, then,

$$\int_0^{2\pi} (\cos mt)(\cos nt)dt = 0 \tag{7.7}$$

since

$$(\cos x)(\cos y) = \frac{1}{2}[\cos(x+y) + \cos(x-y)]$$

The integral of (7.7) can also be confirmed graphically as shown in Figure 7.4, where $m = 2$ and $n = 3$. We observe that the net shaded area above and below the time axis is zero.

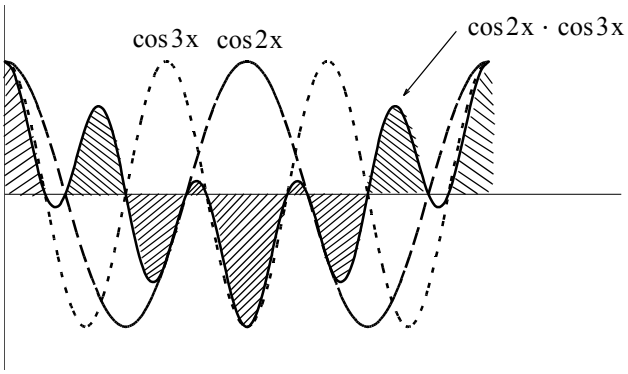


Figure 7.4. *Graphical proof of* $\int_0^{2\pi} (\cos mt)(\cos nt)dt = 0$ *for* $m = 2$ *and* $n = 3$

However, if in (7.6) and (7.7), $m = n$, then,

$$\int_0^{2\pi} (\sin mt)^2 dt = \pi \tag{7.8}$$

and

$$\int_0^{2\pi} (\cos mt)^2 dt = \pi \tag{7.9}$$

The integrals of (7.8) and (7.9) can also be seen to be true graphically with the plots of Figures 7.5 and 7.6.

It was stated earlier that the sine and cosine functions are orthogonal to each other. The simplification obtained by application of the orthogonality properties of the sine and cosine functions, becomes apparent in the discussion that follows.

In (7.1), Page 7–1, for simplicity, we let $\omega = 1$. Then,

$$\begin{aligned} f(t) = \frac{1}{2}a_0 &+ a_1\cos t + a_2\cos 2t + a_3\cos 3t + a_4\cos 4t + \ldots \\ &+ b_1\sin t + b_2\sin 2t + b_3\sin 3t + b_4\sin 4t + \ldots \end{aligned} \tag{7.10}$$

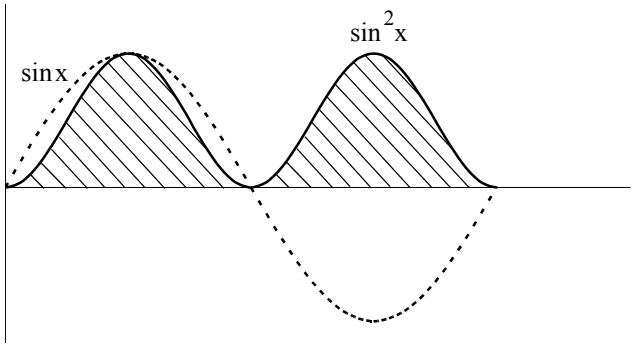


Figure 7.5. *Graphical proof of* $\int_0^{2\pi}(\sin mt)^2 dt = \pi$

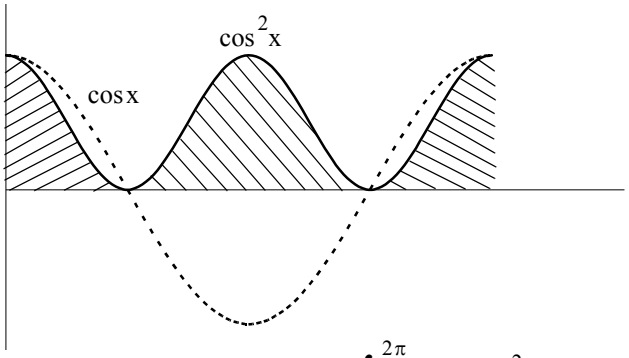


Figure 7.6. *Graphical proof of* $\int_0^{2\pi}(\cos mt)^2 dt = \pi$

To evaluate any coefficient in (7.10), say b_2, we multiply both sides of (7.10) by $\sin 2t$. Then,

$$f(t)\sin 2t = \frac{1}{2}a_0 \sin 2t + a_1 \cos t \sin 2t + a_2 \cos 2t \sin 2t + a_3 \cos 3t \sin 2t + a_4 \cos 4t \sin 2t + \ldots$$

$$b_1 \sin t \sin 2t + b_2 (\sin 2t)^2 + b_3 \sin 3t \sin 2t + b_4 \sin 4t \sin 2t + \ldots$$

Next, we multiply both sides of the above expression by dt, and we integrate over the period 0 to 2π. Then,

$$\int_0^{2\pi} f(t)\sin 2t dt = \frac{1}{2}a_0 \int_0^{2\pi} \sin 2t dt + a_1 \int_0^{2\pi} \cos t \sin 2t dt + a_2 \int_0^{2\pi} \cos 2t \sin 2t dt$$

$$+ a_3 \int_0^{2\pi} \cos 3t \sin 2t dt + \ldots \qquad (7.11)$$

$$+ b_1 \int_0^{2\pi} \sin t \sin 2t dt + b_2 \int_0^{2\pi} (\sin 2t)^2 dt + b_3 \int_0^{2\pi} \sin 3t \sin 2t dt + \ldots$$

We observe that every term on the right side of (7.11) except the term

$$b_2\int_0^{2\pi}(\sin 2t)^2 dt$$

is zero as we found in (7.6) and (7.7). Therefore, (7.11) reduces to

$$\int_0^{2\pi} f(t)\sin 2t\,dt = b_2\int_0^{2\pi}(\sin 2t)^2 dt = b_2\pi$$

or

$$b_2 = \frac{1}{\pi}\int_0^{2\pi} f(t)\sin 2t\,dt$$

and thus we can evaluate this integral for any given function $f(t)$. The remaining coefficients can be evaluated similarly.

The coefficients a_0, a_n, and b_n are found from the following relations.

$$\boxed{\frac{1}{2}a_0 = \frac{1}{2\pi}\int_0^{2\pi} f(t)dt} \tag{7.12}$$

$$\boxed{a_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\cos nt\,dt} \tag{7.13}$$

$$\boxed{b_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\sin nt\,dt} \tag{7.14}$$

The integral of (7.12) yields the average (DC) value of $f(t)$.

7.3 Symmetry in Trigonometric Fourier Series

With a few exceptions such as the waveform of the half–rectified waveform, Page 7–17, the most common waveforms that are used in science and engineering, do not have the average, cosine, and sine terms all present. Some waveforms have cosine terms only, while others have sine terms only. Still other waveforms have or have not DC components. Fortunately, it is possible to predict which terms will be present in the trigonometric Fourier series, by observing whether or not the given waveform possesses some kind of symmetry.

We will discuss three types of symmetry* that can be used to facilitate the computation of the trigonometric Fourier series form. These are:

1. *Odd symmetry* – If a waveform has odd symmetry, that is, if it is an odd function, the series will consist of sine terms only. In other words, if $f(t)$ is an odd function, all the a_i coefficients including a_0, will be zero.

2. *Even symmetry* – If a waveform has even symmetry, that is, if it is an even function, the series will consist of cosine terms only, and a_0 may or may not be zero. In other words, if $f(t)$ is an even function, all the b_i coefficients will be zero.

3. *Half–wave symmetry* – If a waveform has half–wave symmetry (to be defined shortly), only odd (odd cosine and odd sine) harmonics will be present. In other words, all even (even cosine and even sine) harmonics will be zero.

We defined odd and even functions in Chapter 6. We recall that odd functions are those for which

$$-f(-t) = f(t) \tag{7.15}$$

and even functions are those for which

$$f(-t) = f(t) \tag{7.16}$$

Examples of odd and even functions were given in Chapter 6. Generally, an odd function has odd powers of the independent variable t, and an even function has even powers of the independent variable t. Thus, the product of two odd functions or the product of two even functions will result in an even function, whereas the product of an odd function and an even function will result in an odd function. However, the sum (or difference) of an odd and an even function will yield a function which is neither odd nor even.

To understand half–wave symmetry, we recall that any periodic function with period T, is expressed as

$$f(t) = f(t + T) \tag{7.17}$$

that is, the function with value $f(t)$ at any time t, will have the same value again at a later time $t + T$.

A periodic waveform with period T, has half–wave symmetry if

$$-f\,(t + T/2) = f(t) \tag{7.18}$$

---

* *Quartet-wave symmetry is another type of symmetry where a digitally formed waveform with a series of zeros and ones contains only sine odd harmonics. We will not discuss this type of symmetry in this text. For a brief discussion, please refer to Introduction to Simulink with Engineering Applications, Page 7-18, ISBN 978-1-934404-09-6.*

that is, the shape of the negative half–cycle of the waveform is the same as that of the positive half-cycle, but inverted.

We will test the most common waveforms for symmetry in Subsections 7.3.1 through 7.3.5 below.

7.3.1 Symmetry in Square Waveform

For the waveform of Figure 7.7, the average value over one period T is zero, and therefore, $a_0 = 0$. It is also an odd function and has half–wave symmetry since $-f(-t) = f(t)$ and $-f(t + T/2) = f(t)$.

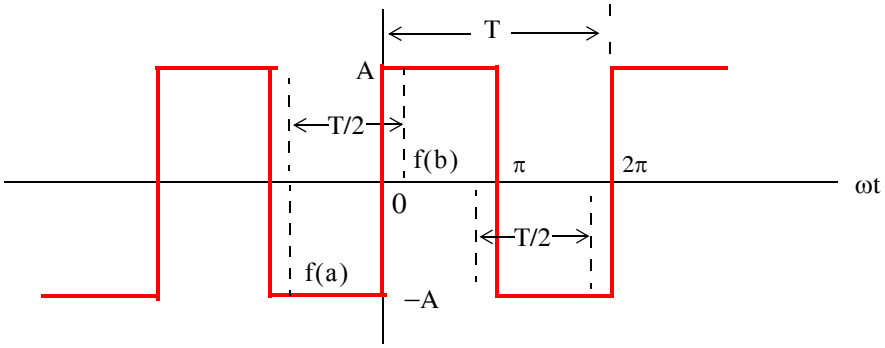


Figure 7.7. Square waveform test for symmetry

An easy method to test for half–wave symmetry is to choose any half–period $T/2$ length on the time axis as shown in Figure 7.7, and observe the values of $f(t)$ at the left and right points on the time axis, such as $f(a)$ and $f(b)$. If there is half–wave symmetry, these will always be equal but will have opposite signs as we slide the half-period $T/2$ length to the left or to the right on the time axis at non–zero values of $f(t)$.

7.3.2 Symmetry in Square Waveform with Ordinate Axis Shifted

If in the square waveform of Figure 7.7 we shift the ordinate axis $\pi/2$ radians to the right, as shown in Figure 7.8, we will observe that the square waveform now becomes an even function, and has half–wave symmetry since $f(-t) = f(t)$ and $-f(t + T/2) = f(t)$. Also, $a_0 = 0$.

Obviously, if the ordinate axis is shifted by any other value other than an odd multiple of $\pi/2$, the waveform will have neither odd nor even symmetry.

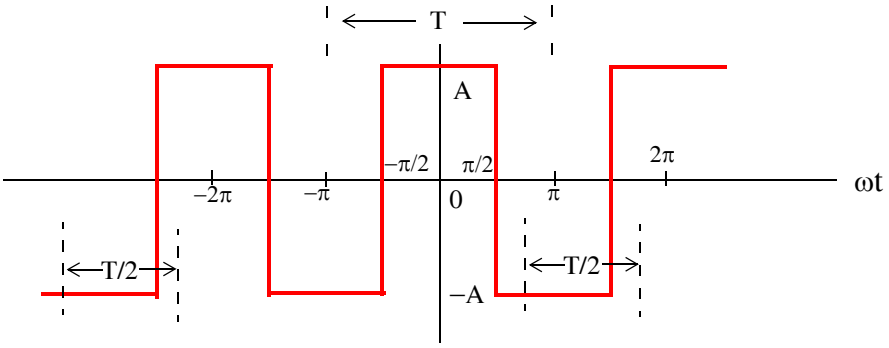


Figure 7.8. Square waveform with ordinate shifted by $\pi/2$

7.3.3 Symmetry in Sawtooth Waveform

For the sawtooth waveform of Figure 7.9, the average value over one period T is zero and therefore, $a_0 = 0$. It is also an odd function because $-f(-t) = f(t)$, but has no half–wave symmetry since $-f(t + T/2) \neq f(t)$

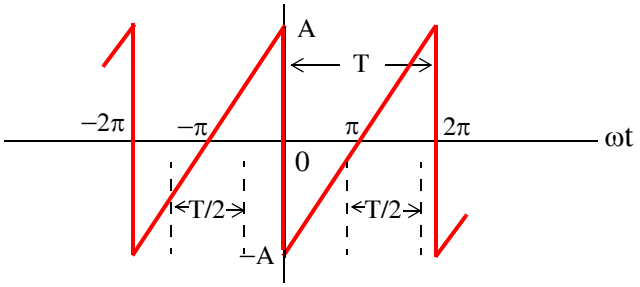


Figure 7.9. Sawtooth waveform test for symmetry

7.3.4 Symmetry in Triangular Waveform

For this triangular waveform of Figure 7.10, the average value over one period T is zero and therefore, $a_0 = 0$. It is also an odd function since $-f(-t) = f(t)$. Moreover, it has half–wave symmetry because $-f(t + T/2) = f(t)$.

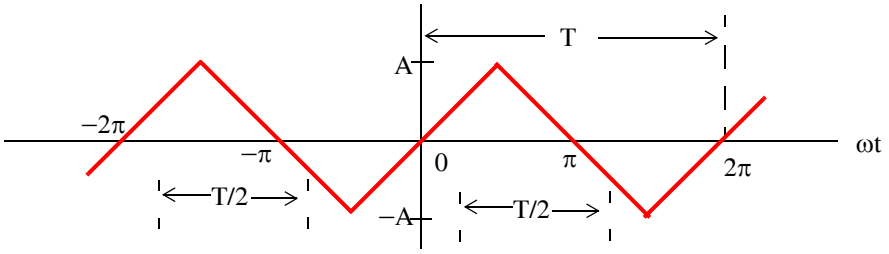


Figure 7.10. Triangular waveform test for symmetry

7.3.5 Symmetry in Fundamental, Second, and Third Harmonics

Figure 7.11 shows a fundamental, second, and third harmonic of a typical sinewave.

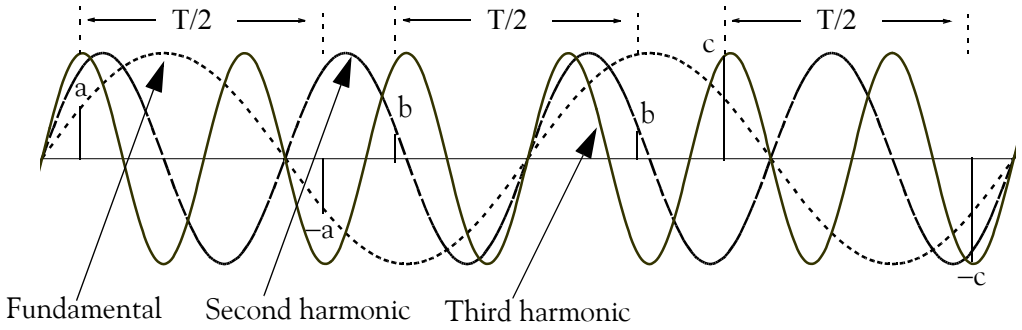


Figure 7.11. Fundamental, second, and third harmonic test for symmetry

In Figure 7.11, the half period $T/2$, is chosen as the half period of the period of the fundamental frequency. This is necessary in order to test the fundamental, second, and third harmonics for half–wave symmetry. The fundamental has half–wave symmetry since the a and $-a$ values, when separated by $T/2$, are equal and opposite. The second harmonic has no half–wave symmetry because the ordinates b on the left and b on the right, although are equal, there are not opposite in sign. The third harmonic has half–wave symmetry since the c and $-c$ values, when separated by $T/2$ are equal and opposite. These waveforms can be either odd or even depending on the position of the ordinate. Also, all three waveforms have zero average value unless the abscissa axis is shifted up or down.

In the expressions of the integrals in (7.12) through (7.14), Page 7–6, the limits of integration for the coefficients a_n and b_n are given as 0 to 2π, that is, one period T. Of course, we can choose the limits of integration as $-\pi$ to $+\pi$. Also, if the given waveform is an odd function, or an even function, or has half–wave symmetry, we can compute the non–zero coefficients a_n and b_n by integrating from 0 to π only, and multiply the integral by 2. Moreover, if the waveform has half–wave symmetry and is also an odd or an even function, we can choose the limits of integration from 0 to $\pi/2$ and multiply the integral by 4. The proof is based on the fact that, the product of two even functions is another even function, and also that the product of two odd functions results also in an even function. However, it is important to remember that when using these shortcuts, we must evaluate the coefficients a_n and b_n for the integer values of n that will result in non–zero coefficients. This point will be illustrated in Subsection 7.4.2, Page 7–14.

7.4 Trigonometric Form of Fourier Series for Common Waveforms

The trigonometric Fourier series of the most common periodic waveforms are derived in Subsections 7.4.1 through 7.4.5 below.

7.4.1 Trigonometric Fourier Series for Square Waveform

For the square waveform of Figure 7.12, the trigonometric series consist of sine terms only because, as we already know from Page 7–8, this waveform is an odd function. Moreover, only odd harmonics will be present since this waveform has also half–wave symmetry. However, we will compute all coefficients to verify this. Also, for brevity, we will assume that $\omega = 1$

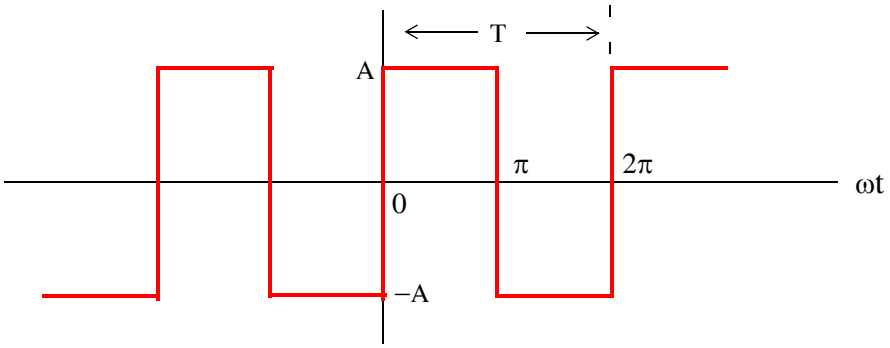


Figure 7.12. Square waveform as odd function

The a_i coefficients are found from

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\cos nt\,dt = \frac{1}{\pi}\left[\int_0^{\pi} A\cos nt\,dt + \int_{\pi}^{2\pi}(-A)\cos nt\,dt\right] = \frac{A}{n\pi}\left(\sin nt\big|_0^{\pi} - \sin nt\big|_{\pi}^{2\pi}\right)$$
$$= \frac{A}{n\pi}(\sin n\pi - 0 - \sin n2\pi + \sin n\pi) = \frac{A}{n\pi}(2\sin n\pi - \sin n2\pi) \tag{7.19}$$

and since n is an integer (positive or negative) or zero, the terms inside the parentheses on the second line of (7.19) are zero and therefore, all a_i coefficients are zero, as expected since the square waveform has odd symmetry. Also, by inspection, the average (DC) value is zero, but if we attempt to verify this using (7.19), we will obtain the indeterminate form $0/0$. To work around this problem, we will evaluate a_0 directly from (7.12), Page 7–6. Thus,

$$a_0 = \frac{1}{\pi}\left[\int_0^{\pi} A\,dt + \int_{\pi}^{2\pi}(-A)\,dt\right] = \frac{A}{\pi}(\pi - 0 - 2\pi + \pi) = 0 \tag{7.20}$$

The b_i coefficients are found from (7.14), Page 7–6, that is,

$$b_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\sin nt\,dt = \frac{1}{\pi}\left[\int_0^{\pi} A\sin nt\,dt + \int_{\pi}^{2\pi}(-A)\sin nt\,dt\right] = \frac{A}{n\pi}\left(-\cos nt\big|_0^{\pi} + \cos nt\big|_{\pi}^{2\pi}\right)$$
$$= \frac{A}{n\pi}(-\cos n\pi + 1 + \cos 2n\pi - \cos n\pi) = \frac{A}{n\pi}(1 - 2\cos n\pi + \cos 2n\pi) \tag{7.21}$$

For n = even, (7.21) yields

$$b_n = \frac{A}{n\pi}(1-2+1) = 0$$

as expected, since the square waveform has half–wave symmetry.

For $n = \text{odd}$, (7.21) reduces to

$$b_n = \frac{A}{n\pi}(1+2+1) = \frac{4A}{n\pi}$$

and thus

$$b_1 = \frac{4A}{\pi}$$

$$b_3 = \frac{4A}{3\pi}$$

$$b_5 = \frac{4A}{5\pi}$$

and so on.

Therefore, the trigonometric Fourier series for the *square waveform with odd symmetry* is

$$\boxed{f(t) = \frac{4A}{\pi}\left(\sin\omega t + \frac{1}{3}\sin 3\omega t + \frac{1}{5}\sin 5\omega t + \ldots\right) = \frac{4A}{\pi}\sum_{n=\text{odd}}\frac{1}{n}\sin n\omega t} \tag{7.22}$$

It was stated above that, if the given waveform has half–wave symmetry, and it is also an odd or an even function, we can integrate from 0 to $\pi/2$, and multiply the integral by 4. This property is verified with the following procedure.

Since the waveform is an odd function and has half–wave symmetry, we are only concerned with the odd b_n coefficients. Then,

$$b_n = 4\frac{1}{\pi}\int_0^{\pi/2} f(t)\sin nt\,dt = \frac{4A}{n\pi}\left(-\cos nt\Big|_0^{\pi/2}\right) = \frac{4A}{n\pi}\left(-\cos n\frac{\pi}{2}+1\right) \tag{7.23}$$

For $n = \text{odd}$, (7.23) becomes

$$b_n = \frac{4A}{n\pi}(-0+1) = \frac{4A}{n\pi} \tag{7.24}$$

as before, and thus the series is as we found earlier.

Next let us consider the square waveform of Figure 7.13 where the ordinate has been shifted to the right by $\pi/2$ radians, and has become an even function. However, it still has half–wave symmetry. Therefore, the trigonometric Fourier series will consist of odd cosine terms only.

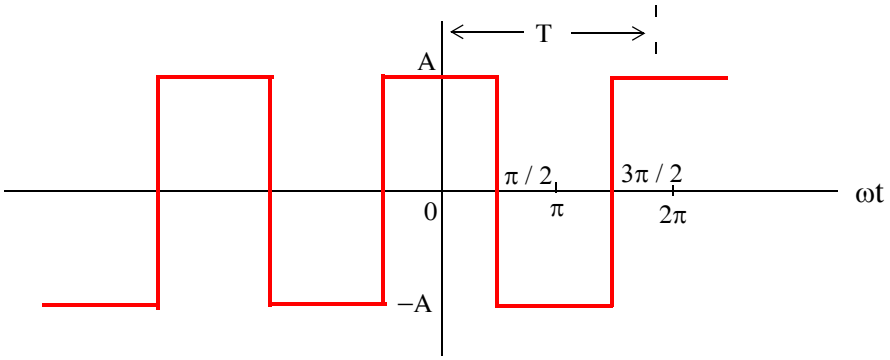


Figure 7.13. Square waveform as even function

Since the waveform has half–wave symmetry and is an even function, it will suffice to integrate from 0 to $\pi/2$, and multiply the integral by 4. The a_n coefficients are found from

$$a_n = 4\frac{1}{\pi}\int_0^{\pi/2} f(t)\cos nt\,dt = \frac{4}{\pi}\left[\int_0^{\pi/2} A\cos nt\,dt\right] = \frac{4A}{n\pi}\left(\sin nt\Big|_0^{\pi/2}\right) = \frac{4A}{n\pi}\left(\sin n\frac{\pi}{2}\right) \tag{7.25}$$

We observe that for $n = \text{even}$, all a_n coefficients are zero, and thus all even harmonics are zero as expected. Also, by inspection, the average (DC) value is zero.

For $n = \text{odd}$, we observe from (7.25) that $\sin n\frac{\pi}{2}$, will alternate between $+1$ and -1 depending on the odd integer assigned to n. Thus,

$$a_n = \pm\frac{4A}{n\pi} \tag{7.26}$$

For $n = 1, 5, 9, 13$, and so on, (7.26) becomes

$$a_n = \frac{4A}{n\pi}$$

and for $n = 3, 7, 11, 15$, and so on, it becomes

$$a_n = \frac{-4A}{n\pi}$$

Then, the trigonometric Fourier series for the *square waveform with even symmetry* is

$$\boxed{f(t) = \frac{4A}{\pi}\left(\cos\omega t - \frac{1}{3}\cos 3\omega t + \frac{1}{5}\cos 5\omega t - \ldots\right) = \frac{4A}{\pi}\sum_{n\,=\,\text{odd}} (-1)^{\frac{(n-1)}{2}}\frac{1}{n}\cos n\omega t} \tag{7.27}$$

The trigonometric series of (7.27) can also be derived as follows:

Since the waveform of Figure 7.12 is the same as that of Figure 7.13, but shifted to the right by $\pi/2$ radians, we can use the relation (7.22), Page 7–12, i.e.,

$$f(t) = \frac{4A}{\pi}\left(\sin\omega t + \frac{1}{3}\sin 3\omega t + \frac{1}{5}\sin 5\omega t + \ldots\right) \tag{7.28}$$

and substitute ωt with $\omega t + \pi/2$, that is, we let $\omega t = \omega\tau + \pi/2$. With this substitution, relation (7.28) becomes

$$\begin{aligned} f(\tau) &= \frac{4A}{\pi}\left[\sin\left(\omega\tau + \frac{\pi}{2}\right) + \frac{1}{3}\sin 3\left(\omega\tau + \frac{\pi}{2}\right) + \frac{1}{5}\sin 5\left(\omega\tau + \frac{\pi}{2}\right) + \ldots\right] \\ &= \frac{4A}{\pi}\left[\sin\left(\omega\tau + \frac{\pi}{2}\right) + \frac{1}{3}\sin\left(3\omega\tau + \frac{3\pi}{2}\right) + \frac{1}{5}\sin\left(5\omega\tau + \frac{5\pi}{2}\right) + \ldots\right] \end{aligned} \tag{7.29}$$

and using the identities $\sin(x + \pi/2) = \cos x$, $\sin(x + 3\pi/2) = -\cos x$, and so on, we rewrite (7.29) as

$$f(\tau) = \frac{4A}{\pi}\left[\cos\omega\tau - \frac{1}{3}\cos 3\omega\tau + \frac{1}{5}\cos 5\omega\tau - \ldots\right] \tag{7.30}$$

and this is the same as (7.27).

Therefore, if we compute the trigonometric Fourier series with reference to one ordinate, and afterwards we want to recompute the series with reference to a different ordinate, we can use the above procedure to save computation time.

7.4.2 Trigonometric Fourier Series for Sawtooth Waveform

The sawtooth waveform of Figure 7.14 is an odd function with no half–wave symmetry; therefore, it contains sine terms only with both odd and even harmonics. Accordingly, we only need to find all b_n coefficients.

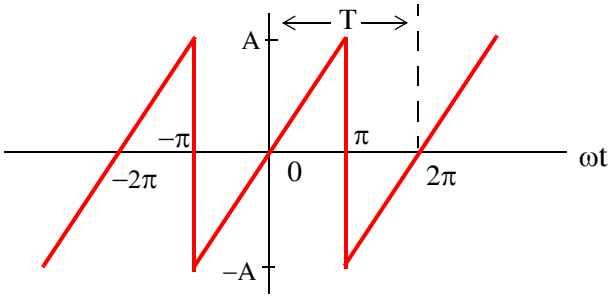


Figure 7.14. Sawtooth waveform

By inspection, the DC component is zero. As before, we will assume that $\omega = 1$.

If we choose the limits of integration from 0 to 2π, we will need to perform two integrations

since

$$f(t) = \begin{cases} \frac{A}{\pi}t & 0 < t < \pi \\ \frac{A}{\pi}t - 2A & \pi < t < 2\pi \end{cases}$$

However, we can choose the limits from $-\pi$ to $+\pi$, and thus we will only need one integration since

$$f(t) = \frac{A}{\pi}t \qquad -\pi < t < \pi$$

Better yet, since the waveform is an odd function, we can integrate from 0 to π, and multiply the integral by 2; this is what we will do.

From tables of integrals,

$$\int x\sin ax\,dx = \frac{1}{a^2}\sin ax - \frac{x}{a}\cos ax \tag{7.31}$$

Then,

$$\begin{aligned} b_n &= \frac{2}{\pi}\int_0^{\pi}\frac{A}{\pi}t\sin nt\,dt = \frac{2A}{\pi^2}\int_0^{\pi}t\sin nt\,dt = \frac{2A}{\pi^2}\left(\frac{1}{n^2}\sin nt - \frac{t}{n}\cos nt\right)\Big|_0^{\pi} \\ &= \frac{2A}{n^2\pi^2}(\sin nt - nt\cos nt)\Big|_0^{\pi} = \frac{2A}{n^2\pi^2}(\sin n\pi - n\pi\cos n\pi) \end{aligned} \tag{7.32}$$

We observe that:

1. If $n = \text{even}$, $\sin n\pi = 0$ and $\cos n\pi = 1$. Then, (7.32) reduces to

$$b_n = \frac{2A}{n^2\pi^2}(-n\pi) = -\frac{2A}{n\pi}$$

that is, the even harmonics have negative coefficients.

2. If $n = \text{odd}$, $\sin n\pi = 0$, $\cos n\pi = -1$. Then,

$$b_n = \frac{2A}{n^2\pi^2}(n\pi) = \frac{2A}{n\pi}$$

that is, the odd harmonics have positive coefficients.

Thus, the trigonometric Fourier series for the *sawtooth waveform with odd symmetry* is

$$\boxed{f(t) = \frac{2A}{\pi}\left(\sin\omega t - \frac{1}{2}\sin 2\omega t + \frac{1}{3}\sin 3\omega t - \frac{1}{4}\sin 4\omega t + \ldots\right) = \frac{2A}{\pi}\sum(-1)^{n-1}\frac{1}{n}\sin n\omega t} \tag{7.33}$$

7.4.3 Trigonometric Fourier Series for Triangular Waveform

The sawtooth waveform of Figure 7.15 is an odd function with half–wave symmetry; then, the trigonometric Fourier series will contain sine terms only with odd harmonics. Accordingly, we only need to evaluate the b_n coefficients. We will choose the limits of integration from 0 to $\pi/2$, and will multiply the integral by 4. As before, we will assume that $\omega = 1$.

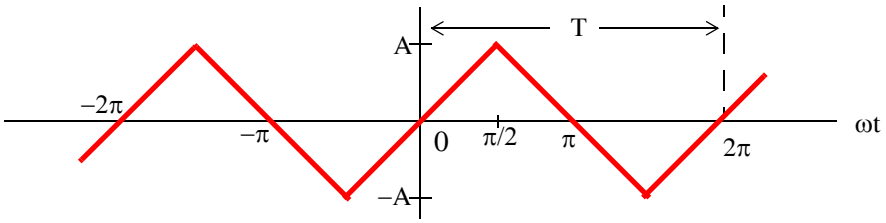


Figure 7.15. Triangular waveform

By inspection, the DC component is zero. From tables of integrals,

$$\int x\sin ax\,dx = \frac{1}{a^2}\sin ax - \frac{x}{a}\cos ax \tag{7.34}$$

Then,

$$\begin{aligned} b_n &= \frac{4}{\pi}\int_0^{\pi/2}\frac{2A}{\pi}t\sin nt\,dt = \frac{8A}{\pi^2}\int_0^{\pi/2} t\sin nt\,dt = \frac{8A}{\pi^2}\left(\frac{1}{n^2}\sin nt - \frac{t}{n}\cos nt\right)\Big|_0^{\pi/2} \\ &= \frac{8A}{n^2\pi^2}(\sin nt - nt\cos nt)\Big|_0^{\pi/2} = \frac{8A}{n^2\pi^2}\left(\sin n\frac{\pi}{2} - n\frac{\pi}{2}\cos n\frac{\pi}{2}\right) \end{aligned} \tag{7.35}$$

We are only interested in the odd integers of n, and we observe that:

$$\cos n\frac{\pi}{2} = 0$$

For odd integers of n, the sine term yields

$$\sin n\frac{\pi}{2} = \begin{cases} 1 \text{ for } n = 1, 5, 9, \ldots \text{ then, } b_n = \dfrac{8A}{n^2\pi^2} \\ -1 \text{ for } n = 3, 7, 11, \ldots \text{ then, } b_n = -\dfrac{8A}{n^2\pi^2} \end{cases}$$

Thus, the trigonometric Fourier series for the *triangular waveform* with odd symmetry is

$$f(t) = \frac{8A}{\pi^2}\left(\sin\omega t - \frac{1}{9}\sin 3\omega t + \frac{1}{25}\sin 5\omega t - \frac{1}{49}\sin 7\omega t + \ldots\right) = \frac{8A}{\pi^2}\sum_{n\,=\,\text{odd}}(-1)^{\frac{(n-1)}{2}}\frac{1}{n^2}\sin n\omega t \tag{7.36}$$

7.4.4 Trigonometric Fourier Series for Half–Wave Rectifier Waveform

The circuit of Figure 7.16 is a *half–wave rectifier* whose input is the sinusoid $v_{in}(t) = \sin\omega t$, and its output $v_{out}(t)$ is defined as

$$v_{out}(t) = \begin{cases} \sin\omega t & 0 < \omega t < \pi \\ 0 & \pi < \omega t < 2\pi \end{cases} \tag{7.37}$$

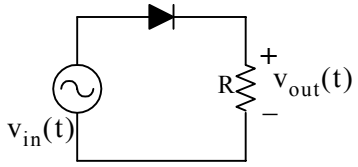


Figure 7.16. Circuit for half–wave rectifier

We will express $v_{out}(t)$ as a trigonometric Fourier series, and we will assume that $\omega = 1$. The input and output waveforms are shown in Figures 7.17 and 7.18 respectively.

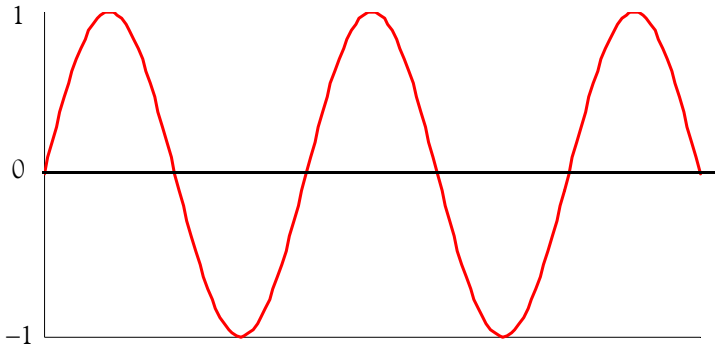


Figure 7.17. Input $v_{in}(t)$ for the circuit of Figure 7.16

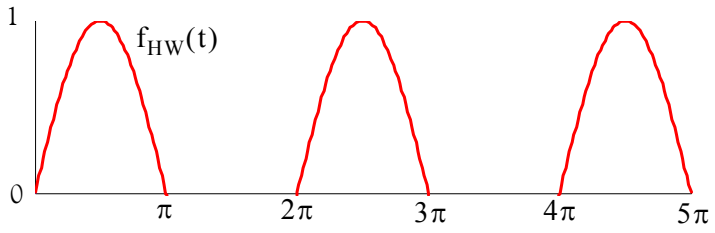


Figure 7.18. Output $v_{out}(t)$ for the circuit of Figure 7.16

We choose the ordinate at point 0 as shown in Figure 7.19.

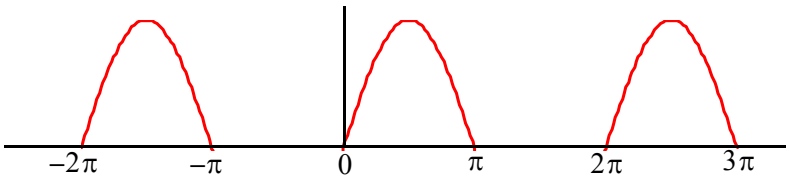


Figure 7.19. Half–wave rectifier waveform for the circuit of Figure 7.16

By inspection, the average is a non–zero value, and the waveform has neither odd nor even symmetry. Therefore, we expect all terms to be present.

The a_n coefficients are found from

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\cos nt\,dt$$

or

$$a_n = \frac{A}{\pi}\int_0^{\pi} \sin t\cos nt\,dt + \frac{A}{\pi}\int_{\pi}^{2\pi} 0\cos nt\,dt$$

and from tables of integrals

$$\int(\sin mx)(\cos nx)dx = -\frac{\cos(m-n)x}{2(m-n)} - \frac{\cos(m+n)x}{2(m+n)} \quad (m^2 \neq n^2)$$

Then,

$$\begin{aligned} a_n &= \frac{A}{\pi}\left\{-\frac{1}{2}\left[\frac{\cos(1-n)t}{1-n} + \frac{\cos(1+n)t}{1+n}\right]\Bigg|_0^{\pi}\right\} \\ &= -\frac{A}{2\pi}\left\{\left[\frac{\cos(\pi-n\pi)}{1-n} + \frac{\cos(\pi+n\pi)}{1+n}\right] - \left[\frac{1}{1-n} + \frac{1}{n+1}\right]\right\} \end{aligned} \tag{7.38}$$

Using the trigonometric identities

$$\cos(x-y) = \cos x\cos y + \sin x\sin y$$

and

$$\cos(x+y) = \cos x\cos y - \sin x\sin y$$

we obtain

$$\cos(\pi - n\pi) = \cos\pi\cos n\pi + \sin\pi\sin n\pi = -\cos n\pi$$

and

$$\cos(\pi + n\pi) = \cos\pi\cos n\pi - \sin\pi\sin n\pi = -\cos n\pi$$

Then, by substitution into (7.38),

$$a_n = -\frac{A}{2\pi}\left\{\left[\frac{-\cos n\pi}{1-n} + \frac{-\cos n\pi}{1+n}\right] - \frac{2}{1-n^2}\right\} = \frac{A}{2\pi}\left\{\left[\frac{\cos n\pi}{1-n} + \frac{\cos n\pi}{1+n}\right] + \frac{2}{1-n^2}\right\}$$

or

$$a_n = \frac{A}{2\pi}\left(\frac{\cos n\pi + n\cos n\pi + \cos n\pi - n\cos n\pi}{1-n^2} + \frac{2}{1-n^2}\right) = \frac{A}{\pi}\left(\frac{\cos n\pi + 1}{(1-n^2)}\right) \quad n \neq 1 \tag{7.39}$$

Now, we can evaluate all the a_n coefficients, except a_1, from (7.39).

First, we will evaluate a_0 to obtain the DC value. By substitution of $n = 0$, we obtain

$$a_0 = 2A/\pi$$

Therefore, the DC value is

$$\frac{1}{2}a_0 = \frac{A}{\pi} \tag{7.40}$$

We cannot use (7.39) to obtain the value of a_1 because this relation is not valid for $n = 1$; therefore, we will evaluate the integral

$$a_1 = \frac{A}{\pi}\int_0^{\pi} \sin t \cos t\, dt$$

From tables of integrals,

$$\int (\sin ax)(\cos ax)dx = \frac{1}{2a}(\sin ax)^2$$

and thus,

$$a_1 = \frac{A}{2\pi}(\sin t)^2\Big|_0^{\pi} = 0 \tag{7.41}$$

From (7.39) with $n = 2, 3, 4, 5, \ldots$, we obtain

$$a_2 = \frac{A}{\pi}\left(\frac{\cos 2\pi + 1}{(1-2^2)}\right) = -\frac{2A}{3\pi} \tag{7.42}$$

$$a_3 = \frac{A(\cos 3\pi + 1)}{\pi(1-3^2)} = 0 \tag{7.43}$$

We see that for odd integers of n, $a_n = 0$. However, for $n = \text{even}$, we obtain

$$a_4 = \frac{A(\cos 4\pi + 1)}{\pi(1-4^2)} = -\frac{2A}{15\pi} \tag{7.44}$$

$$a_6 = \frac{A(\cos 6\pi + 1)}{\pi(1-6^2)} = -\frac{2A}{35\pi} \tag{7.45}$$

$$a_8 = \frac{A(\cos 8\pi + 1)}{\pi(1-8^2)} = -\frac{2A}{63\pi} \tag{7.46}$$

and so on.

Next, we need to evaluate the b_n coefficients. For this waveform,

$$b_n = A\frac{1}{\pi}\int_0^{2\pi} f(t)\sin nt\,dt = \frac{A}{\pi}\int_0^{\pi} \sin t \sin nt\,dt + \frac{A}{\pi}\int_{\pi}^{2\pi} 0 \sin nt\,dt$$

and from tables of integrals,

$$\int(\sin mx)(\sin nx)dx = \frac{\sin(m-n)x}{2(m-n)} - \frac{\sin(m+n)x}{2(m+n)} \quad (m^2 \neq n^2)$$

Therefore,

$$b_n = \frac{A}{\pi}\cdot\frac{1}{2}\left\{\left[\frac{\sin(1-n)t}{1-n} - \frac{\sin(1+n)t}{1+n}\right]\Bigg|_0^{\pi}\right\}$$

$$= \frac{A}{2\pi}\left[\frac{\sin(1-n)\pi}{1-n} - \frac{\sin(1+n)\pi}{1+n} - 0 + 0\right] = 0 \quad (n \neq 1)$$

that is, all the b_n coefficients, except b_1, are zero.

We will find b_1 by direct substitution into (7.14), Page 7–6, for $n = 1$. Thus,

$$b_1 = \frac{A}{\pi}\int_0^{\pi}(\sin t)^2 dt = \frac{A}{\pi}\left[\frac{t}{2} - \frac{\sin 2t}{4}\right]\Bigg|_0^{\pi} = \frac{A}{\pi}\left[\frac{\pi}{2} - \frac{\sin 2\pi}{4}\right] = \frac{A}{2} \tag{7.47}$$

Combining (7.40), with (7.42) through (7.47), we find that the trigonometric Fourier series for the *half–wave rectifier with no symmetry* is

$$\boxed{f(t) = \frac{A}{\pi} + \frac{A}{2}\sin t - \frac{A}{\pi}\left[\frac{\cos 2t}{3} + \frac{\cos 4t}{15} + \frac{\cos 6t}{35} + \frac{\cos 8t}{63} + \ldots\right]} \tag{7.48}$$

7.4.5 Trigonometric Fourier Series for Full–Wave Rectifier Waveform

Figure 7.20 shows a *full–wave rectifier* circuit with input the sinusoid $v_{in}(t) = A\sin\omega t$. The output of that circuit is $v_{out}(t) = |A\sin\omega t|$.

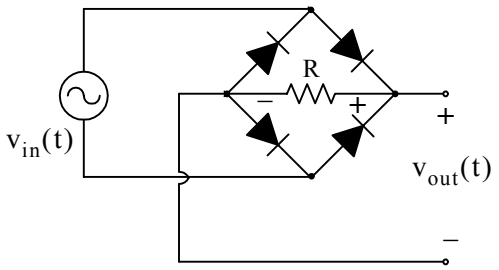


Figure 7.20. Full-wave rectifier circuit

The input and output waveforms are shown in Figures 7.21 and 7.22 respectively. We will express $v_{out}(t)$ as a trigonometric Fourier series, and we will assume that $\omega = 1$.

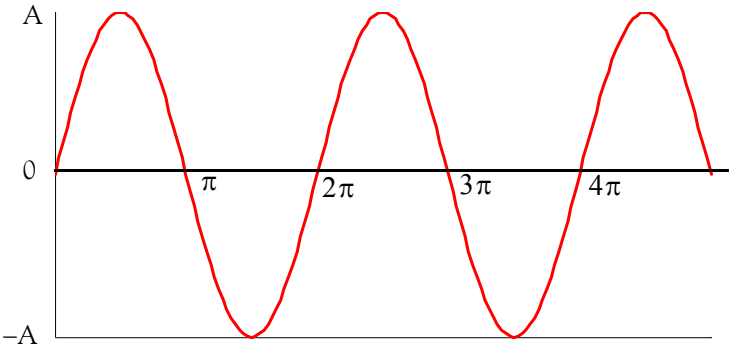


Figure 7.21. Input sinusoid for the full–rectifier circuit of Figure 7.20

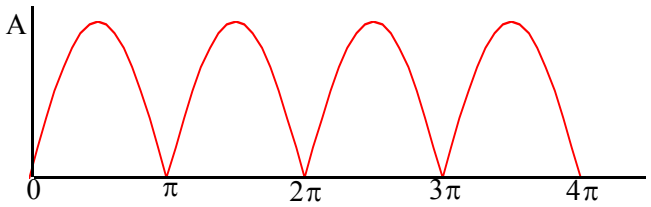


Figure 7.22. Output waveform for full–rectifier circuit of Figure 7.20

We choose the ordinate as shown in Figure 7.23.

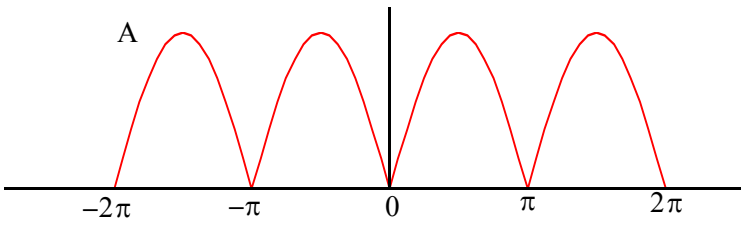


Figure 7.23. Full–wave rectified waveform with even symmetry

By inspection, the average is a non–zero value. We choose the period of the input sinusoid so that the output will be expressed in terms of the fundamental frequency. We also choose the limits of integration as $-\pi$ and $+\pi$, we observe that the waveform has even symmetry. Therefore, we expect only cosine terms to be present.

The a_n coefficients are found from

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\cos nt\,dt$$

where for this waveform,

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} A\sin t\cos nt\,dt = \frac{2A}{\pi}\int_0^{\pi} \sin t\cos nt\,dt \tag{7.49}$$

and from tables of integrals,

$$\int(\sin mx)(\cos nx)dx = \frac{\cos(m-n)x}{2(n-m)} - \frac{\cos(m+n)x}{2(m+n)} \quad (m^2 \neq n^2)$$

Since

$$\cos(x-y) = \cos(y-x) = \cos x\cos y + \sin x\sin y$$

we express (7.49) as

$$\begin{aligned} a_n &= \frac{2A}{\pi}\cdot\frac{1}{2}\left\{\left[\frac{\cos(n-1)t}{n-1} - \frac{\cos(n+1)t}{n+1}\right]\Bigg|_0^{\pi}\right\} \\ &= \frac{A}{\pi}\left\{\left[\frac{\cos(n-1)\pi}{n-1} - \frac{\cos(n+1)\pi}{n+1}\right] - \left[\frac{1}{n-1} - \frac{1}{n+1}\right]\right\} \\ &= \frac{A}{\pi}\left[\frac{1-\cos(n\pi+\pi)}{n+1} + \frac{\cos(n\pi-\pi)-1}{n-1}\right] \end{aligned} \tag{7.50}$$

To simplify the last expression in (7.50), we make use of the trigonometric identities

$$\cos(n\pi+\pi) = \cos n\pi\cos\pi - \sin n\pi\sin\pi = -\cos n\pi$$

and

$$\cos(n\pi-\pi) = \cos n\pi\cos\pi + \sin n\pi\sin\pi = -\cos n\pi$$

Then, (7.50) simplifies to

$$\begin{aligned} a_n &= \frac{A}{\pi}\left[\frac{1+\cos n\pi}{n+1} - \frac{1+\cos n\pi}{n-1}\right] = \frac{A}{\pi}\left[\frac{-2+(n-1)\cos n\pi-(n+1)\cos n\pi}{n^2-1}\right] \\ &= \frac{-2A(\cos n\pi+1)}{\pi(n^2-1)} \quad n \neq 1 \end{aligned} \tag{7.51}$$

Now, we can evaluate all the a_n coefficients, except a_1, from (7.51). First, we will evaluate a_0 to obtain the DC value. By substitution of $n = 0$, we obtain

$$a_0 = \frac{4A}{\pi}$$

Therefore, the DC value is

$$\frac{1}{2}a_0 = \frac{2A}{\pi} \tag{7.52}$$

From (7.51) we observe that for all $n = odd$, other than $n = 1$, $a_n = 0$.

To obtain the value of a_1, we must evaluate the integral

$$a_1 = \frac{1}{\pi}\int_0^{\pi} \sin t \cos t dt$$

From tables of integrals,

$$\int (\sin ax)(\cos ax)dx = \frac{1}{2a}(\sin ax)^2$$

and thus,

$$a_1 = \frac{1}{2\pi}(\sin t)^2\Big|_0^{\pi} = 0 \tag{7.53}$$

For $n = even$, from (7.51) we obtain

$$a_2 = \frac{-2A(\cos 2\pi + 1)}{\pi(2^2 - 1)} = -\frac{4A}{3\pi} \tag{7.54}$$

$$a_4 = \frac{-2A(\cos 4\pi + 1)}{\pi(4^2 - 1)} = -\frac{4A}{15\pi} \tag{7.55}$$

$$a_6 = \frac{-2A(\cos 6\pi + 1)}{\pi(6^2 - 1)} = -\frac{4A}{35\pi} \tag{7.56}$$

$$a_8 = \frac{-2A(\cos 8\pi + 1)}{\pi(8^2 - 1)} = -\frac{4A}{63\pi} \tag{7.57}$$

and so on. Then, combining the terms of (7.52) with (7.54) through (7.57) we obtain

$$f(t) = \frac{2A}{\pi} - \frac{4A}{\pi}\left\{\frac{\cos 2\omega t}{3} + \frac{\cos 4\omega t}{15} + \frac{\cos 6\omega t}{35} + \frac{\cos 8\omega t}{63} + \ldots\right\} \tag{7.58}$$

Therefore, the trigonometric form of the Fourier series for the *full-wave rectifier with even symmetry* is

$$f(t) = \frac{2A}{\pi} - \frac{4A}{\pi} \sum_{n = 2, 4, 6, \ldots}^{\infty} \frac{1}{(n^2 - 1)} \cos n\omega t \tag{7.59}$$

This series of (7.59) shows that there is no component of the input (fundamental) frequency. This is because we chose the period to be from $-\pi$ and $+\pi$. Generally, the period is defined as the shortest period of repetition. In any waveform where the period is chosen appropriately, it is very unlikely that a Fourier series will consist of even harmonic terms only.

7.5 Gibbs Phenomenon

In Subsection 7.4.1, Page 7–12, we found that the trigonometric form of the Fourier series of the square waveform is

$$f(t) = \frac{4A}{\pi}\left(\sin\omega t + \frac{1}{3}\sin 3\omega t + \frac{1}{5}\sin 5\omega t + \ldots\right) = \frac{4A}{\pi} \sum_{n = \text{odd}} \frac{1}{n} \sin n\omega t$$

Figure 7.24 shows the first 11 harmonics and their sum. As we add more and more harmonics, the sum looks more and more like the square waveform. However, the crests do not become flattened; this is known as *Gibbs phenomenon* and it occurs because of the discontinuity of the perfect square waveform as it changes from $+A$ to $-A$.

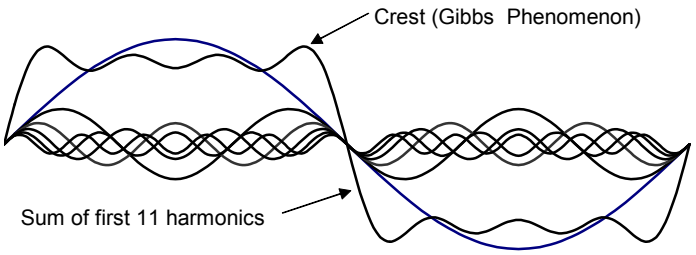


Figure 7.24. Gibbs phenomenon

7.6 Alternate Forms of the Trigonometric Fourier Series

We recall that the trigonometric Fourier series is expressed as

$$\begin{aligned} f(t) = {} & \frac{1}{2}a_0 + a_1\cos\omega t + a_2\cos 2\omega t + a_3\cos 3\omega t + a_4\cos 4\omega t + \ldots \\ & + b_1\sin\omega t + b_2\sin 2\omega t + b_3\sin 3\omega t + b_4\sin 4\omega t + \ldots \end{aligned} \tag{7.60}$$

If a given waveform does not have any kind of symmetry, it may be advantageous of using the *alternate form of the trigonometric Fourier series* where the cosine and sine terms of the same frequency are grouped together, and the sum is combined to a single term, either cosine or sine. However, we still need to compute the a_n and b_n coefficients separately.

For the derivation of the alternate forms, we will use the triangle shown in Figure 7.25.

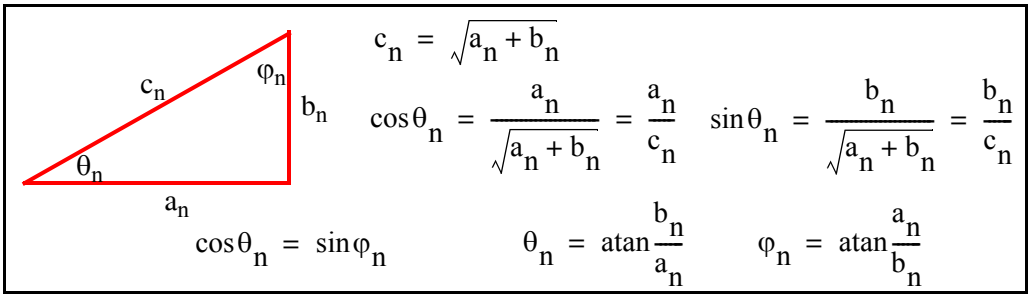


Figure 7.25. Derivation of the alternate form of the trigonometric Fourier series

We assume $\omega = 1$, and for $n = 1, 2, 3, \ldots$, we rewrite (7.60) as

$$\begin{aligned} f(t) &= \frac{1}{2}a_0 + c_1\left(\frac{a_1}{c_1}\cos t + \frac{b_1}{c_1}\sin t\right) + c_2\left(\frac{a_2}{c_2}\cos 2t + \frac{b_2}{c_2}\sin 2t\right) + \ldots \\ &\quad + c_n\left(\frac{a_n}{c_n}\cos nt + \frac{b_n}{c_n}\sin nt\right) \\ &= \frac{1}{2}a_0 + c_1\left(\underbrace{\cos\theta_1\cos t + \sin\theta_1\sin t}_{\cos(t-\theta_1)}\right) + c_2\left(\underbrace{\cos\theta_2\cos 2t + \sin\theta_2\sin 2t}_{\cos(2t-\theta_2)}\right) + \ldots \\ &\quad + c_n\left(\underbrace{\cos\theta_n\cos nt + \sin\theta_n\sin nt}_{\cos(nt-\theta_n)}\right) \end{aligned}$$

and, in general, for $\omega \neq 1$, we obtain

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} c_n\cos(n\omega t - \theta_n) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} c_n\cos\left(n\omega t - \text{atan}\frac{b_n}{a_n}\right) \tag{7.61}$$

Similarly,

$$\begin{aligned} f(t) &= \frac{1}{2}a_0 + c_1\left(\underbrace{\sin\varphi_1\cos t + \cos\varphi_1\sin t}_{\sin(t+\varphi_1)}\right) \\ &\quad c_2\left(\underbrace{\sin\varphi_2\cos 2t + \cos\varphi_2\sin 2t}_{\sin(2t+\varphi_2)}\right) + \ldots + c_n\left(\underbrace{\sin\varphi_n\cos nt + \cos\varphi_n\sin nt}_{\sin(nt+\varphi_n)}\right) \end{aligned}$$

and, in general, where $\omega \neq 1$, we obtain

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} c_n \sin(n\omega t + \varphi_n) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} c_n \sin\left(n\omega t + \operatorname{atan}\frac{a_n}{b_n}\right) \tag{7.62}$$

When used in circuit analysis, (7.61) and (7.62) can be expressed as phasors. Since it is customary to use the cosine function in the time domain to phasor transformation, we choose to use the transformation of (7.63) below.

$$\frac{1}{2}a_0 + \sum_{n=1}^{\infty} c_n \cos\left(n\omega t - \operatorname{atan}\frac{b_n}{a_n}\right) \Leftrightarrow \frac{1}{2}a_0 + \sum_{n=1}^{\infty} c_n \angle -\operatorname{atan}\frac{b_n}{a_n} \tag{7.63}$$

Example 7.1

Find the first 5 terms of the alternate form of the trigonometric Fourier series for the waveform of Figure 7.26.

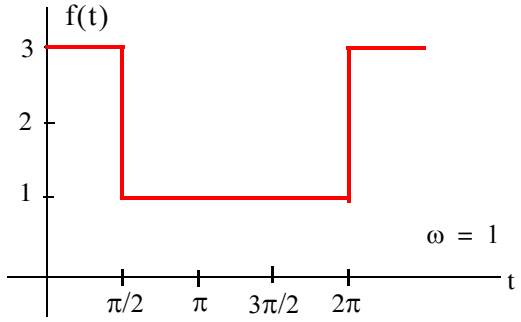


Figure 7.26. Waveform for Example 7.1

Solution:

The given waveform has no symmetry; thus, we expect both cosine and sine functions with odd and even terms present. Also, by inspection the DC value is not zero.

We will compute the a_n and b_n coefficients, the DC value, and we will combine them to obtain an expression in the form of (7.63). Then,

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_0^{\pi/2}(3)\cos nt\,dt + \frac{1}{\pi}\int_{\pi/2}^{2\pi}(1)\cos nt\,dt = \frac{3}{n\pi}\sin nt\Big|_0^{\pi/2} + \frac{1}{n\pi}\sin nt\Big|_{\pi/2}^{2\pi} \\ &= \frac{3}{n\pi}\sin n\frac{\pi}{2} + \frac{1}{n\pi}\sin n2\pi - \frac{1}{n\pi}\sin n\frac{\pi}{2} = \frac{2}{n\pi}\sin n\frac{\pi}{2} \end{aligned} \tag{7.64}$$

We observe that for $n = \text{even}$, $a_n = 0$.

For $n = \text{odd}$,

$$a_1 = \frac{2}{\pi} \tag{7.65}$$

and

$$a_3 = -\frac{2}{3\pi} \tag{7.66}$$

The DC value is

$$\begin{aligned}\frac{1}{2}a_0 &= \frac{1}{2\pi}\int_0^{\pi/2}(3)dt + \frac{1}{2\pi}\int_{\pi/2}^{2\pi}(1)dt = \frac{1}{2\pi}(3t\big|_0^{\pi/2} + t\big|_{\pi/2}^{2\pi}) \\ &= \frac{1}{2\pi}\left(\frac{3\pi}{2} + 2\pi - \frac{\pi}{2}\right) = \frac{1}{2\pi}(\pi + 2\pi) = \frac{3}{2}\end{aligned} \tag{7.67}$$

The b_n coefficients are

$$\begin{aligned}b_n &= \frac{1}{\pi}\int_0^{\pi/2}(3)\sin nt\,dt + \frac{1}{\pi}\int_{\pi/2}^{2\pi}(1)\sin nt\,dt = \frac{-3}{n\pi}\cos nt\Big|_0^{\pi/2} + \frac{-1}{n\pi}\cos nt\Big|_{\pi/2}^{2\pi} \\ &= \frac{-3}{n\pi}\cos n\frac{\pi}{2} + \frac{3}{n\pi} + \frac{-1}{n\pi}\cos n2\pi + \frac{1}{n\pi}\cos n\frac{\pi}{2} = \frac{1}{n\pi}(3 - \cos n2\pi) = \frac{2}{n\pi}\end{aligned} \tag{7.68}$$

Then,

$$b_1 = 2/\pi \tag{7.69}$$

$$b_2 = 1/\pi \tag{7.70}$$

$$b_3 = 2/3\pi \tag{7.71}$$

$$b_4 = 1/2\pi \tag{7.72}$$

From (7.63),

$$\frac{1}{2}a_0 + \sum_{n=1}^{\infty} c_n \cos\left(n\omega t - \text{atan}\frac{b_n}{a_n}\right) \Leftrightarrow \frac{1}{2}a_0 + \sum_{n=1}^{\infty} c_n \angle -\text{atan}\frac{b_n}{a_n}$$

where

$$c_n\angle -\text{atan}\frac{b_n}{a_n} = \sqrt{a_n^2 + b_n^2}\angle -\text{atan}\frac{b_n}{a_n} = \sqrt{a_n^2 + b_n^2}\angle -\theta_n = a_n - jb_n \tag{7.73}$$

Thus, for $n = 1, 2, 3,$ and 4, we obtain:

$$\begin{aligned}a_1 - jb_1 &= \frac{2}{\pi} - j\frac{2}{\pi} = \sqrt{\left(\frac{2}{\pi}\right)^2 + \left(\frac{2}{\pi}\right)^2}\angle -45° \\ &= \sqrt{\frac{8}{\pi^2}}\angle -45° = \frac{2\sqrt{2}}{\pi}\angle -45° \Leftrightarrow \frac{2\sqrt{2}}{\pi}\cos(\omega t - 45°)\end{aligned} \tag{7.74}$$

Similarly,

$$a_2 - jb_2 = 0 - j\frac{1}{\pi} = \frac{1}{\pi}\angle -90° \Leftrightarrow \frac{1}{\pi}\cos(2\omega t - 90°) \tag{7.75}$$

$$a_3 - jb_3 = -\frac{2}{3\pi} - j\frac{2}{3\pi} = \frac{2\sqrt{2}}{3\pi}\angle -135° \Leftrightarrow \frac{2\sqrt{2}}{3\pi}\cos(3\omega t - 135°) \tag{7.76}$$

and

$$a_4 - jb_4 = 0 - j\frac{1}{2\pi} = \frac{1}{2\pi}\angle -90° \Leftrightarrow \frac{1}{2\pi}\cos(4\omega t - 90°) \tag{7.77}$$

Combining the terms of (7.67) with (7.74) through (7.77), we find that the alternate form of the trigonometric Fourier series representing the waveform of this example is

$$\begin{aligned} f(t) = \frac{3}{2} + \frac{1}{\pi}\Big[& 2\sqrt{2}\cos(\omega t - 45°) + \cos(2\omega t - 90°) \\ & + \frac{2\sqrt{2}}{3}\cos(3\omega t - 135°) + \frac{1}{2}\cos(4\omega t - 90°) + \ldots \Big] \end{aligned} \tag{7.78}$$

7.7 Circuit Analysis with Trigonometric Fourier Series

When the excitation of an electric circuit is a non–sinusoidal waveform such as those we presented thus far, we can use Fouries series to determine the response of a circuit. The procedure is illustrated with the examples that follow.

Example 7.2

The input to the series RC circuit of Figure 7.27, is the square waveform of Figure 7.28. Compute the voltage $v_C(t)$ across the capacitor. Consider only the first three terms of the series, and assume $\omega = 1$.

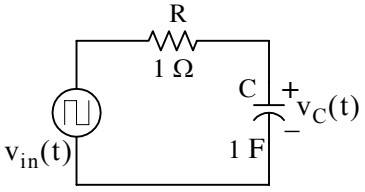


Figure 7.27. Circuit for Example 7.2

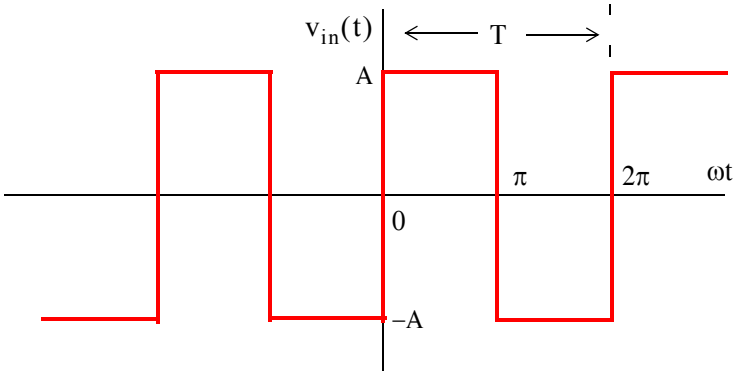


Figure 7.28. Input waveform for the circuit of Figure 7.27

Solution:

In Subsection 7.4.1, Page 7–11, we found that the waveform of Figure 7.28 can be represented by the trigonometric Fourier series as

$$f(t) = \frac{4A}{\pi}\left(\sin\omega t + \frac{1}{3}\sin 3\omega t + \frac{1}{5}\sin 5\omega t + \ldots\right) \tag{7.79}$$

Since this series is the sum of sinusoids, we will use phasor analysis to obtain the solution. The equivalent phasor circuit is shown in Figure 7.29.

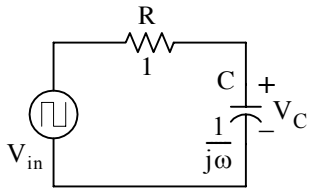


Figure 7.29. Phasor circuit for Example 7.2

We let n represent the number of terms in the Fourier series. For this example, we are only interested in the first three odd terms, that is, $n = 1, 3, \text{and } 5$.

By the voltage division expression,

$$V_{Cn} = \frac{1/(jn\omega)}{1 + 1/(jn\omega)}V_{inn} = \frac{1}{1 + jn}V_{inn} \tag{7.80}$$

With reference to (7.79) the phasors of the first 3 odd terms of (7.80) are

$$\frac{4A}{\pi}\sin t = \frac{4A}{\pi}\cos(t - 90°) \Leftrightarrow V_{in1} = \frac{4A}{\pi}\angle -90° \tag{7.81}$$

$$\frac{4A}{\pi}\cdot\frac{1}{3}\sin 3t = \frac{4A}{\pi}\cdot\frac{1}{3}\cos(3t - 90°) \Leftrightarrow V_{in3} = \frac{4A}{\pi}\cdot\frac{1}{3}\angle -90° \tag{7.82}$$

$$\frac{4A}{\pi}\cdot\frac{1}{5}\sin 5t = \frac{4A}{\pi}\cdot\frac{1}{5}\cos(5t-90°) \Leftrightarrow V_{in5} = \frac{4A}{\pi}\cdot\frac{1}{5}\angle -90° \tag{7.83}$$

By substitution of (7.81) through (7.83) into (7.80), we obtain the phasor and time domain voltages indicated in (7.84) through (7.86) below.

$$\begin{aligned} V_{C1} &= \frac{1}{1+j}\cdot\frac{4A}{\pi}\angle -90° = \frac{1}{\sqrt{2}\angle 45°}\cdot\frac{4A}{\pi}\angle -90° \\ &= \frac{4A}{\pi}\cdot\frac{\sqrt{2}}{2}\angle -135° \Leftrightarrow \frac{4A}{\pi}\cdot\frac{\sqrt{2}}{2}\cos(t-135°) \end{aligned} \tag{7.84}$$

$$\begin{aligned} V_{C3} &= \frac{1}{1+j3}\cdot\frac{4A}{\pi}\cdot\frac{1}{3}\angle -90° = \frac{1}{\sqrt{10}\angle 71.6°}\frac{4A}{\pi}\cdot\frac{1}{3}\angle -90° \\ &= \frac{4A}{\pi}\cdot\frac{\sqrt{10}}{30}\angle -161.6° \Leftrightarrow \frac{4A}{\pi}\cdot\frac{\sqrt{10}}{30}\cos(3t-161.6°) \end{aligned} \tag{7.85}$$

$$\begin{aligned} V_{C5} &= \frac{1}{1+j5}\cdot\frac{4A}{\pi}\cdot\frac{1}{5}\angle -90° = \frac{1}{\sqrt{26}\angle 78.7°}\frac{4A}{\pi}\cdot\frac{1}{5}\angle -90° \\ &= \frac{4A}{\pi}\cdot\frac{\sqrt{26}}{130}\angle -168.7° \Leftrightarrow \frac{4A}{\pi}\cdot\frac{\sqrt{26}}{130}\cos(5t-168.7°) \end{aligned} \tag{7.86}$$

Thus, the capacitor voltage in the time domain is

$$v_C(t) = \frac{4A}{\pi}\left[\frac{\sqrt{2}}{2}\cos(t-135°)+\frac{\sqrt{10}}{30}\cos(3t-161.6°)+\frac{\sqrt{26}}{130}\cos(5t-168.7°)+\ldots\right] \tag{7.87}$$

Assuming that in the circuit of Figure 7.27 the capacitor is initially discharged, we expect that capacitor voltage will consist of alternating rising and decaying exponentials. Let us plot relation (7.87) using the MATLAB script below assuming that $A = 1$.

```
t=0:pi/64:4*pi; Vc=(4./pi).*((sqrt(2)./2).*cos(t–135.*pi./180)+...
(sqrt(10)./30).*cos(3.*t–161.6.*pi./180)+(sqrt(26)./130).*cos(5.*t–168.7.*pi./180)); plot(t,Vc)
```

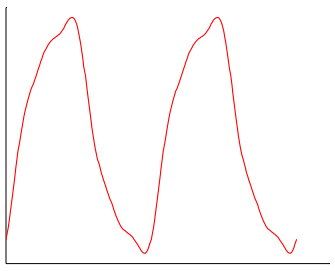


Figure 7.30. Waveform for relation (7.87)

The waveform of Figure 7.30 is a rudimentary presentation of the capacitor voltage for the circuit of Figure 7.27. However, it will improve if we add a sufficient number of harmonics in (7.87).

We can obtain a more accurate waveform for the capacitor voltage of Figure 7.27 with the Simulink model of Figure 7.31.

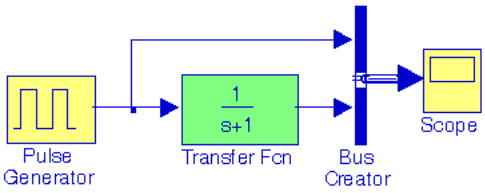


Figure 7.31. Simulink model for the circuit of Figure 7.27

The input and output waveforms are shown in Figure 7.32.

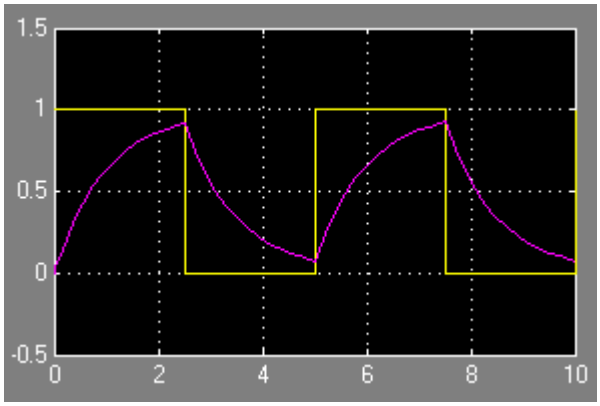


Figure 7.32. Input and output waveforms for the model of Figure 7.31

7.8 The Exponential Form of the Fourier Series

The Fourier series are often expressed in exponential form. The advantage of the exponential form is that we only need to perform one integration rather than two, one for the a_n, and another for the b_n coefficients in the trigonometric form of the series. Moreover, in most cases the integration is simpler.

The exponential form is derived from the trigonometric form by substitution of

$$\cos\omega t = \frac{e^{j\omega t} + e^{-j\omega t}}{2} \tag{7.88}$$

$$\sin\omega t = \frac{e^{j\omega t} - e^{-j\omega t}}{j2} \tag{7.89}$$

into f(t). Thus,

$$f(t) = \frac{1}{2}a_0 + a_1\left(\frac{e^{j\omega t} + e^{-j\omega t}}{2}\right) + a_2\left(\frac{e^{j2\omega t} + e^{-j2\omega t}}{2}\right) + \ldots + b_1\left(\frac{e^{j\omega t} - e^{-j\omega t}}{j2}\right) + b_2\left(\frac{e^{j2\omega t} - e^{-j2\omega t}}{j2}\right) + \ldots \tag{7.90}$$

and grouping terms with same exponents, we obtain

$$f(t) = \ldots + \left(\frac{a_2}{2} - \frac{b_2}{j2}\right)e^{-j2\omega t} + \left(\frac{a_1}{2} - \frac{b_1}{j2}\right)e^{-j\omega t} + \frac{1}{2}a_0 + \left(\frac{a_1}{2} + \frac{b_1}{j2}\right)e^{j\omega t} + \left(\frac{a_2}{2} + \frac{b_2}{j2}\right)e^{j2\omega t} \tag{7.91}$$

The terms of (7.91) in parentheses are usually denoted as

$$C_{-n} = \frac{1}{2}\left(a_n - \frac{b_n}{j}\right) = \frac{1}{2}(a_n + jb_n) \tag{7.92}$$

$$C_n = \frac{1}{2}\left(a_n + \frac{b_n}{j}\right) = \frac{1}{2}(a_n - jb_n) \tag{7.93}$$

$$C_0 = \frac{1}{2}a_0 \tag{7.94}$$

Then, (7.91) is written as

$$\boxed{f(t) = \ldots + C_{-2}e^{-j2\omega t} + C_{-1}e^{-j\omega t} + C_0 + C_1e^{j\omega t} + C_2e^{j2\omega t} + \ldots} \tag{7.95}$$

We must remember that the C_i coefficients, except C_0, are complex and occur in complex conjugate pairs, that is,

$$C_{-n} = C_n^* \tag{7.96}$$

We can derive a general expression for the complex coefficients C_n, by multiplying both sides of (7.95) by $e^{-jn\omega t}$ and integrating over one period, as we did in the derivation of the a_n and b_n coefficients of the trigonometric form. Then, with $\omega = 1$,

$$\int_0^{2\pi} f(t)e^{-jnt}dt = \ldots + \int_0^{2\pi} C_{-2}e^{-j2t}e^{-jnt}dt + \int_0^{2\pi} C_{-1}e^{-jt}e^{-jnt}dt + \int_0^{2\pi} C_0e^{-jnt}dt + \int_0^{2\pi} C_1e^{jt}e^{-jnt}dt + \int_0^{2\pi} C_2e^{j2t}e^{-jnt}dt + \ldots + \int_0^{2\pi} C_ne^{jnt}e^{-jnt}dt \tag{7.97}$$

We observe that all the integrals on the right side of (7.96) are zero except the last. Therefore,

$$\int_0^{2\pi} f(t)e^{-jnt}dt = \int_0^{2\pi} C_n e^{jnt}e^{-jnt}dt = \int_0^{2\pi} C_n dt = 2\pi C_n$$

or

$$C_n = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-jnt}dt \tag{7.98}$$

and, in general, for $\omega \neq 1$,

$$\boxed{C_n = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-jn\omega t}d(\omega t)} \tag{7.99}$$

or

$$\boxed{C_n = \frac{1}{T}\int_0^{T} f(t)e^{-jn\omega t}d(\omega t)} \tag{7.100}$$

We can derive the trigonometric Fourier series from the exponential series by addition and subtraction of the exponential form coefficients C_n and C_{-n}. Thus, from (7.92) and (7.93),

$$C_n + C_{-n} = \frac{1}{2}(a_n - jb_n + a_n + jb_n)$$

or

$$\boxed{a_n = C_n + C_{-n}} \tag{7.101}$$

Similarly,

$$C_n - C_{-n} = \frac{1}{2}(a_n - jb_n - a_n - jb_n) \tag{7.102}$$

or

$$\boxed{b_n = j(C_n - C_{-n})} \tag{7.103}$$

7.9 Symmetry in Exponential Fourier Series

Since the coefficients of the Fourier series in exponential form appear as complex numbers, we can use the properties in Subsections 7.9.1 through 7.9.5 below to determine the symmetry in the exponential Fourier series.

7.9.1 Even Functions

For even functions, all coefficients C_i are real.

We recall from (7.92) and (7.93) that

$$C_{-n} = \frac{1}{2}\left(a_n - \frac{b_n}{j}\right) = \frac{1}{2}(a_n + jb_n) \tag{7.104}$$

and

$$C_n = \frac{1}{2}\left(a_n + \frac{b_n}{j}\right) = \frac{1}{2}(a_n - jb_n) \tag{7.105}$$

Since even functions have no sine terms, the b_n coefficients in (7.104) and (7.105) are zero. Therefore, both C_{-n} and C_n are real.

7.9.2 Odd Functions

For odd functions, all coefficients C_i are imaginary.

Since odd functions have no cosine terms, the a_n coefficients in (7.104) and (7.105) are zero. Therefore, both C_{-n} and C_n are imaginary.

7.9.3 Half–Wave Symmetry

If there is half–wave symmetry, $C_n = 0$ for $n = \text{even}$.

We recall from the trigonometric Fourier series that if there is half–wave symmetry, all even harmonics are zero. Therefore, in (7.104) and (7.105) the coefficients a_n and b_n are both zero for $n = \text{even}$, and thus, both C_{-n} and C_n are also zero for $n = \text{even}$.

7.9.4 No Symmetry

If there is no symmetry, $f(t)$ is complex.

This is evident from Subparagraphs 7.9.1 and 7.9.2, and relations (7.104) and (7.105).

7.9.5 Relation of C_{-n} to C_n^*

$C_{-n} = C_n^*$ *always.*

This is evident from (7.104) and (7.105).

Example 7.3

Compute the exponential Fourier series for the square waveform of Figure 7.33 below. Assume that $\omega = 1$.

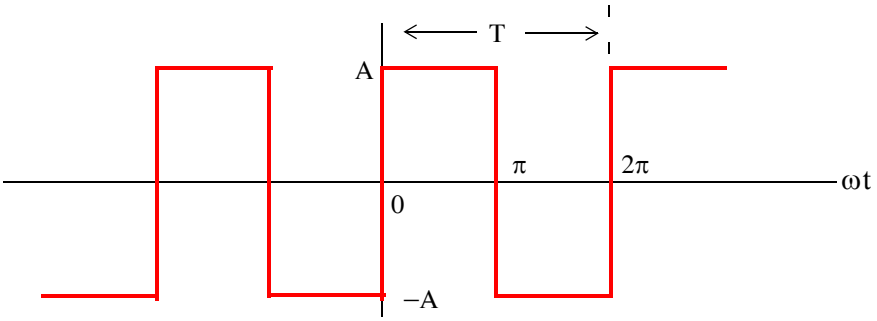


Figure 7.33. Waveform for Example 7.3

Solution:

This is the same waveform as in Subsection 7.4.1, Page 7–11, and as we know, it is an odd function, has half–wave symmetry, and its DC component is zero. Therefore, the C_n coefficients will be imaginary, $C_n = 0$ for $n = \text{even}$, and $C_0 = 0$. Using (7.99), Page 7–33, with $\omega = 1$, we obtain

$$C_n = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-jnt}dt = \frac{1}{2\pi}\int_0^{\pi} Ae^{-jnt}dt + \frac{1}{2\pi}\int_{\pi}^{2\pi} -Ae^{-jnt}dt$$

and for $n = 0$,

$$C_0 = \frac{1}{2\pi}\left[\int_0^{\pi} Ae^{-0}dt + \int_{\pi}^{2\pi}(-A)e^{-0}dt\right] = \frac{A}{2\pi}(\pi - 2\pi + \pi) = 0$$

as expected.

For $n \neq 0$,

$$\begin{aligned} C_n &= \frac{1}{2\pi}\left[\int_0^{\pi} Ae^{-jnt}dt + \int_{\pi}^{2\pi} -Ae^{-jnt}dt\right] = \frac{1}{2\pi}\left[\frac{A}{-jn}e^{-jnt}\Big|_0^{\pi} + \frac{-A}{-jn}e^{-jnt}\Big|_{\pi}^{2\pi}\right] \\ &= \frac{1}{2\pi}\left[\frac{A}{-jn}(e^{-jn\pi} - 1) + \frac{A}{jn}(e^{-jn2\pi} - e^{-jn\pi})\right] = \frac{A}{2j\pi n}(1 - e^{-jn\pi} + e^{-jn2\pi} - e^{-jn\pi}) \\ &= \frac{A}{2j\pi n}(1 + e^{-jn2\pi} - 2e^{-jn\pi}) = \frac{A}{2j\pi n}(e^{-jn\pi} - 1)^2 \end{aligned} \tag{7.106}$$

For $n = \text{even}$, $e^{-jn\pi} = 1$; then,

$$C_{\substack{n \\ n = \text{even}}} = \frac{A}{2j\pi n}(e^{-jn\pi} - 1)^2 = \frac{A}{2j\pi n}(1 - 1)^2 = 0 \tag{7.107}$$

as expected.

For $n = \text{odd}$, $e^{-jn\pi} = -1$. Therefore,

$$C_{\substack{n\\ n=\text{odd}}} = \frac{A}{2j\pi n}(e^{-jn\pi}-1)^2 = \frac{A}{2j\pi n}(-1-1)^2 = \frac{A}{2j\pi n}(-2)^2 = \frac{2A}{j\pi n} \tag{7.108}$$

Using (7.94), Page 7-32, that is,

$$f(t) = \ldots + C_{-2}e^{-j2\omega t} + C_{-1}e^{-j\omega t} + C_0 + C_1 e^{\,j\omega t} + C_2 e^{\,j2\omega t} + \ldots$$

we obtain the *exponential Fourier series for the square waveform with odd symmetry* as

$$f(t) = \frac{2A}{j\pi}\left(\ldots - \frac{1}{3}e^{-j3\omega t} - e^{-j\omega t} + e^{\,j\omega t} + \frac{1}{3}e^{\,j3\omega t}\right) = \frac{2A}{j\pi}\sum_{n=\text{odd}}\frac{1}{n}e^{\,jn\omega t} \tag{7.109}$$

The minus (–) sign of the first two terms within the parentheses results from the fact that $C_{-n} = C_n^*$. For instance, since $C_3 = 2A/j3\pi$, it follows that $C_{-3} = C_3^* = -2A/j3\pi$. We observe that $f(t)$ is purely imaginary, as expected, since the waveform is an odd function.

To prove that (7.109) above and (7.22), Page 7–12 are the same, we group the two terms inside the parentheses of (7.109) for which $n = 1$; this will produce the fundamental frequency $\sin\omega t$. Then, we group the two terms for which $n = 3$, and this will produce the third harmonic $\sin 3\omega t$, and so on.

7.10 Line Spectra

When the Fourier series are known, it is useful to plot the amplitudes of the harmonics on a frequency scale that shows the first (fundamental frequency) harmonic, and the higher harmonics times the amplitude of the fundamental. Such a plot is known as *line spectrum* and shows the spectral lines that would be displayed by a *spectrum analyzer**.

Figure 7.34 shows the line spectrum of the square waveform in Subsection 7.4.1, Page 7–11.

Figure 7.34. Line spectrum for square waveform in Subsection 7.4.1

Figure 7.35 shows the line spectrum for the half–wave rectifier in Subsection 7.4.4, Page 7–18.

* *An instrument that displays the spectral lines of a waveform.*

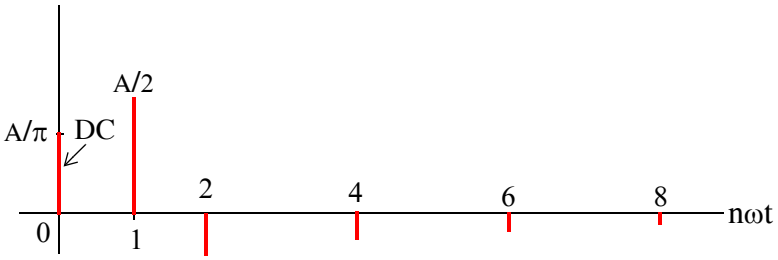


Figure 7.35. Line spectrum for half–wave rectifier, Page 7–17

The line spectra of other waveforms can be easily constructed from the Fourier series.

Example 7.4

Compute the exponential Fourier series for the waveform of Figure 7.36, and plot its line spectra. Assume $\omega = 1$.

Solution:

This recurrent rectangular pulse is used extensively in digital communications systems. To determine how faithfully such pulses will be transmitted, it is necessary to know the frequency components.

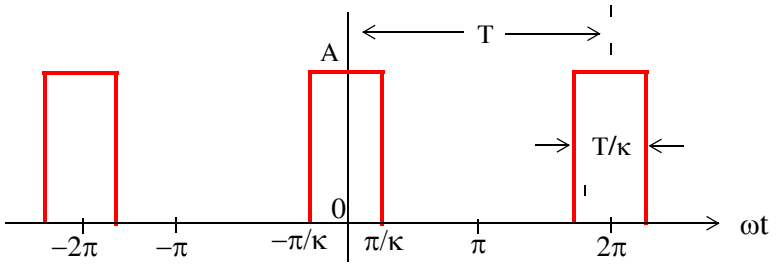


Figure 7.36. Waveform for Example 7.4

As shown in Figure 7.36, the pulse duration is T/k. Thus, the recurrence interval (period) T, is k times the pulse duration. In other words, k is the ratio of the pulse repetition time to the duration of each pulse. For this example, the components of the exponential Fourier series are found from

$$C_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} Ae^{-jnt}dt = \frac{A}{2\pi}\int_{-\pi/k}^{\pi/k} e^{-jnt}dt \tag{7.110}$$

The value of the average (DC component) is found by letting $n = 0$. Then, from (7.110) we obtain

$$C_0 = \frac{A}{2\pi}t\Big|_{-\pi/k}^{\pi/k} = \frac{A}{2\pi}\left(\frac{\pi}{k}+\frac{\pi}{k}\right)$$

or

$$C_0 = \frac{A}{k} \tag{7.111}$$

For the values for $n \neq 0$, integration of (7.110) yields

$$C_n = \frac{A}{-jn2\pi} e^{-jnt}\Big|_{-\pi/k}^{\pi/k} = \frac{A}{n\pi} \cdot \frac{e^{jn\pi/k} - e^{-jn\pi/k}}{j2} = \frac{A}{n\pi} \cdot \sin\left(\frac{n\pi}{k}\right) = A\frac{\sin(n\pi/k)}{n\pi}$$

or

$$C_n = \frac{A}{k} \cdot \frac{\sin(n\pi/k)}{n\pi/k} \tag{7.112}$$

and thus,

$$\boxed{f(t) = \sum_{n=-\infty}^{\infty} \frac{A}{k} \cdot \frac{\sin(n\pi/k)}{n\pi/k}} \tag{7.113}$$

The relation of (7.113) has the $\sin x/x$ form, and the line spectra are shown in Figures 7.37 through 7.39, for $k = 2$, $k = 5$ and $k = 10$ respectively by using the MATLAB scripts below.

```
fplot('sin(2.*x)./(2.*x)',[–4  4  –0.4  1.2])
fplot('sin(5.*x)./(5.*x)',[–4  4  –0.4  1.2])
fplot('sin(10.*x)./(10.*x)',[–4  4 –0.4  1.2])
```

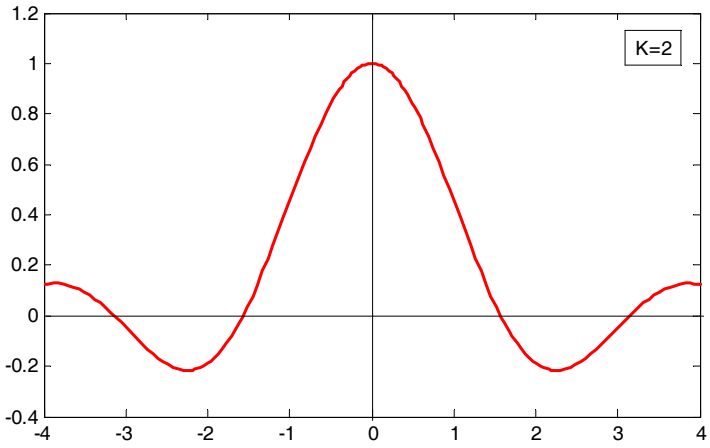


Figure 7.37. Line spectrum of (7.113) for $k = 2$

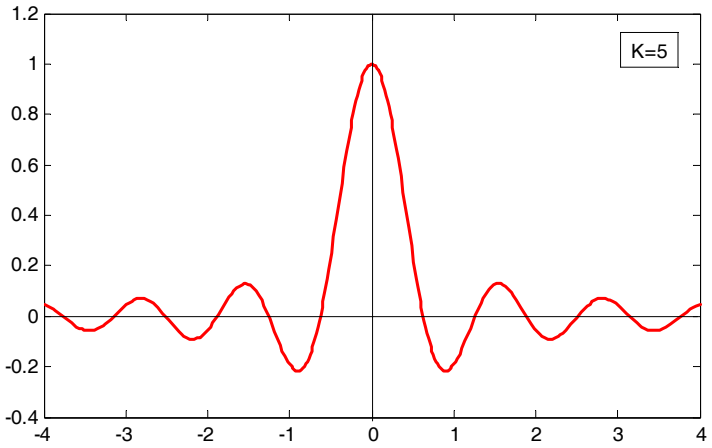


Figure 7.38. Line spectrum of (7.113) for k = 5

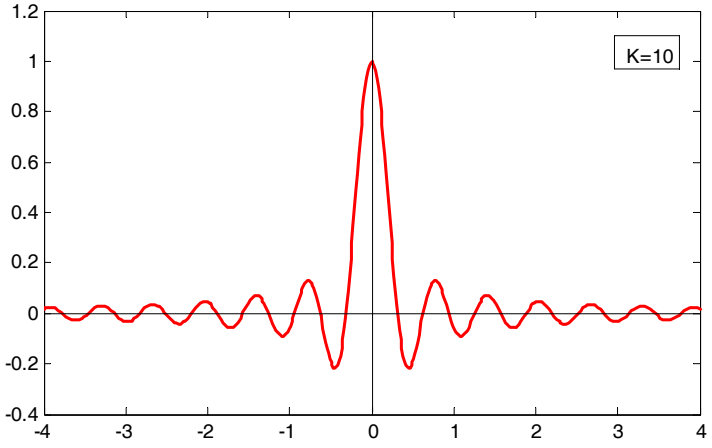


Figure 7.39. Line spectrum of (7.113) for k = 10

The spectral lines are separated by the distance $1/k$ and thus, as k becomes larger, the lines get closer together while the lines are further apart as k gets smaller.

Example 7.5

Use the result of Example 7.4 to compute the exponential Fourier series of the unit impulse train $A\delta(t \pm 2\pi n)$ shown in Figure 7.40.

Solution:

From relation (7.112), Page 7–38,

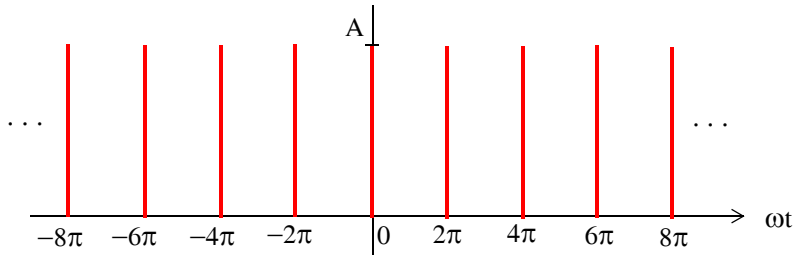


Figure 7.40. Impulse train for Example 7.4

$$C_n = \frac{A}{k} \cdot \frac{\sin(n\pi/k)}{n\pi/k} \tag{7.114}$$

and the pulse width was defined as T/k, that is,

$$\frac{T}{k} = \frac{2\pi}{k} \tag{7.115}$$

Next, let us represent the impulse train of Figure 7.40, as a recurrent pulse with amplitude

$$A = \frac{1}{T/k} = \frac{1}{2\pi/k} = \frac{k}{2\pi} \tag{7.116}$$

as shown in Figure 7.41.

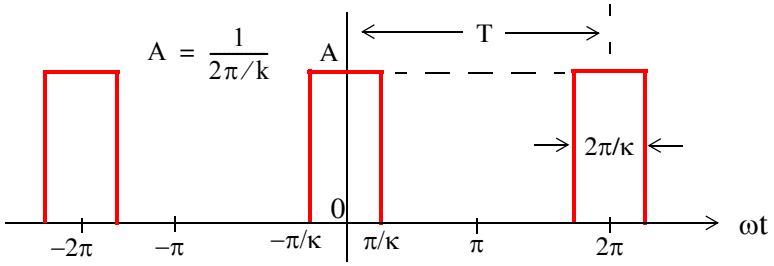


Figure 7.41. Recurrent pulse with amplitude $A = \frac{1}{2\pi/k}$

By substitution of (7.116) into (7.114), we obtain

$$C_n = \frac{k/2\pi}{k} \cdot \frac{\sin(n\pi/k)}{n\pi/k} = \frac{1}{2\pi}\frac{\sin(n\pi/k)}{n\pi/k} \tag{7.117}$$

and as $\pi/k \to 0$, we observe from Figure 7.41, that each recurrent pulse becomes a unit impulse, and the total number of the pulses reduce to a unit impulse train. Moreover, recalling that $\lim_{x \to 0} \frac{\sin x}{x} = 1$, we see that (7.117) reduces to $C_n = \frac{1}{2\pi}$, that is, all coefficients of the exponential Fourier series have the same amplitude and thus,

$$f(t) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} e^{jn\omega t} \tag{7.118}$$

The series of (7.118) reveals that the line spectrum of the impulse train of Figure 7.40, consists of a train of equal amplitude, and are equally spaced harmonics as shown in Figure 7.42.

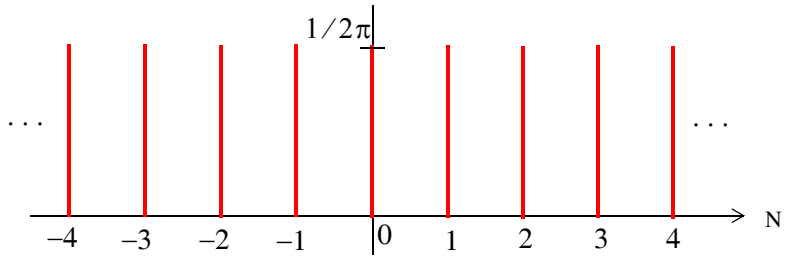


Figure 7.42. Line spectrum for Example 7.5

Since these spectral lines extend from $-\infty$ to $+\infty$, the bandwidth approaches infinity.

Let us reconsider the train of recurrent pulses shown in Figure 7.43.

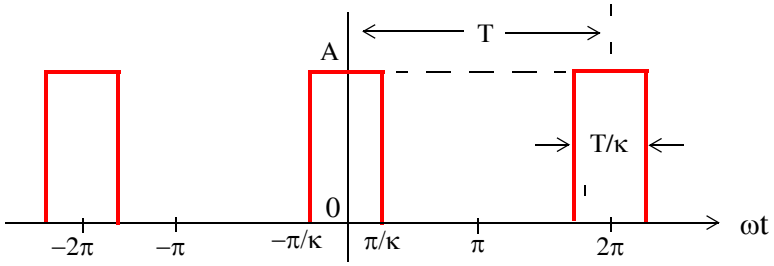


Figure 7.43. Recurrent pulse with $T \to \infty$

Now, let us suppose that the pulses to the left and right of the pulse centered around zero, become less and less frequent; or in other words, the period T approaches infinity. In this case, there is only one pulse left (the one centered around zero). As $T \to \infty$, the fundamental frequency approaches zero, that is, $\omega \to 0$ as T approaches infinity. Accordingly, the frequency difference between consecutive harmonics becomes smaller. In this case, the lines in the line spectrum come closer together, and the line spectrum becomes a continuous spectrum. This forms the basis of the Fourier transform that we will study in the next chapter.

7.11 Computation of RMS Values from Fourier Series

The RMS value of a waveform consisting of sinusoids of different frequencies, is equal to the square root of the sum of the squares of the RMS values of each sinusoid. Thus, if

$$i = I_0 + I_1\cos(\omega_1 t \pm \theta_1) + I_2\cos(\omega_2 t \pm \theta_2) + \ldots + I_N\cos(\omega_N t \pm \theta_N) \tag{7.119}$$

where I_0 represents a constant current, and $I_1, I_2, \ldots, I_N$ represent the amplitudes of the sinusoids, the RMS value of i is found from

$$\boxed{I_{RMS} = \sqrt{I_0^2 + I_{1\,RMS}^2 + I_{2\,RMS}^2 + \ldots + I_{N\,RMS}^2}} \tag{7.120}$$

or

$$\boxed{I_{RMS} = \sqrt{I_0^2 + \frac{1}{2}I_{1\,m}^2 + \frac{1}{2}I_{2\,m}^2 + \ldots + \frac{1}{2}I_{N\,m}^2}} \tag{7.121}$$

The proof of (7.120) is based on Parseval's theorem; we will state this theorem in the next chapter. A brief description of the proof of (7.120) follows.

We recall that the RMS (effective) value of a function, such as current $i(t)$, is defined as

$$I_{RMS} = \sqrt{\frac{1}{T}\int_0^T i^2 dt} \tag{7.122}$$

Substitution of (7.119) into (7.122), will produce the terms I_0^2, $I_{1m}^2[\cos(\omega_1 t - \theta_1)]^2$, and other similar terms representing higher order harmonics. The result will also contain products of cosine functions multiplied by a constant, or other cosine terms of different harmonic frequencies. But as we know, from the orthogonality principle, the integration of (7.122), will produce all zero terms except the cosine squared terms which, for each harmonic, will be

$$\frac{I_m^2\frac{T}{2}}{T} = \frac{1}{2}I_m^2 \tag{7.123}$$

as in (7.121).

Example 7.6

Consider the waveform of Figure 7.44.

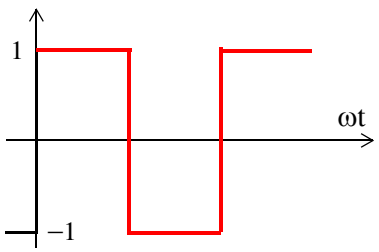


Figure 7.44. Waveform for Example 7.6

Find the I_{RMS} value of this waveform by application of

a. relation (7.122)

b. relation (7.121)

Solution:

a. By inspection, the period is $T = 2\pi$ as shown in Figure 7.45.

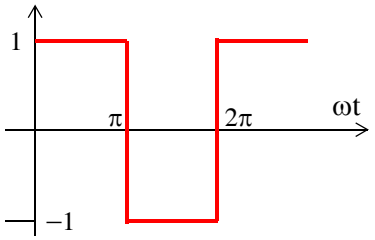


Figure 7.45. Waveform of Example 7.6 showing period $T = 2\pi$

Then,

$$I_{RMS}^2 = \frac{1}{T}\int_0^T i^2 dt = \frac{1}{2\pi}\int_0^{2\pi} i^2 d(\omega t) = \frac{1}{2\pi}\left[\int_0^{\pi} 1^2 d(\omega t) + \int_{\pi}^{2\pi} (-1)^2 d(\omega t)\right]$$

$$= \frac{1}{2\pi}\left[\omega t\big|_0^{\pi} + \omega t\big|_{\pi}^{2\pi}\right] = \frac{1}{2\pi}[2\pi] = 1$$

or

$$I_{RMS} = 1$$

b. In Subsection 7.4.1, Page 7–11, we found that the given waveform may be written as

$$i(t) = \frac{4}{\pi}\left(\sin\omega t + \frac{1}{3}\sin 3\omega t + \frac{1}{5}\sin 5\omega t + \ldots\right) \tag{7.124}$$

and as we know, the RMS value of a sinusoid is a real number independent of the frequency and the phase angle, and it is equal to 0.707 times its maximum value, that is, $I_{RMS} = 0.707 I_{max}$. Then, from (7.121) and (7.124),

$$I_{RMS} = \frac{4}{\pi}\sqrt{0 + \frac{1}{2}(1)^2 + \frac{1}{2}\left(\frac{1}{3}\right)^2 + \frac{1}{2}\left(\frac{1}{5}\right)^2 + \ldots} = 0.97 \tag{7.125}$$

This is a good approximation to unity, considering that higher harmonics have been neglected.

7.12 Computation of Average Power from Fourier Series

We can compute the average power of a Fourier series from the relation

$$\begin{aligned} P_{ave} &= P_{dc} + P_{1ave} + P_{2ave} + \ldots \\ &= V_{dc}I_{dc} + V_{1RMS}I_{1RMS}\cos\theta_1 + V_{2RMS}I_{2RMS}\cos\theta_2 + \ldots \end{aligned} \tag{7.126}$$

The proof is obtained from the definition of average power, i.e.,

$$P_{ave} = \frac{1}{T}\int_0^T pdt = \frac{1}{T}\int_0^T vidt \tag{7.127}$$

and the expression for the alternate trigonometric Fourier series, that is,

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} c_n \cos(n\omega t - \theta_n) \tag{7.128}$$

where $f(t)$ can represent voltages and currents. Then, by substitution of these series for v and i into (7.127), we will find that the products of v and i that have different frequencies, will be zero, and only the products of the same frequency terms will have non-zero values. The non–zero values will represent the average power for each harmonic in (7.126).

Example 7.7

For the circuit of Figure 7.46, compute:

a. The current $i_c(t)$ given that $v_{in}(t) = 6\left(\cos\omega t - \frac{1}{3}\cos 3\omega t\right)$ V where $\omega = 1000$ r/s.

b. The average power P_{ave} delivered by the voltage source.

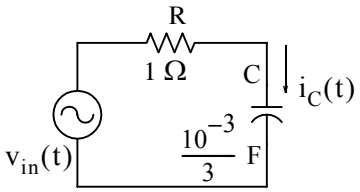


Figure 7.46. Circuit for Example 7.7

Solution:

a. We will use the subscripts 1 and 3 to represent the quantities due to the fundamental and third harmonic frequencies respectively. Since the excitation consists of two sinusoids of different frequencies, we can use phasor quantities, and we will denote them with capital letters.

Then,

$$v_{in1}(t) = 6\cos\omega t \Leftrightarrow V_{in1} = 6\angle 0°\ \text{V}$$

$$\frac{-j}{\omega_1 C} = \frac{-j}{10^3 \times 10^{-3}/3} = -j3$$

$$Z_1 = 1 - j3 = \sqrt{10}\angle -71.6°$$

$$I_{C1} = \frac{V_{in1}}{Z_1} = \frac{6\angle 0°}{\sqrt{10}\angle -71.6°} = 1.90\angle 71.6° \Leftrightarrow i_{C1}(t) = 1.90\cos(\omega t + 71.6°)\ \text{A} \tag{7.129}$$

Next,

$$v_{in3}(t) = -2\cos 3\omega t = 2\cos(3\omega t + 180°) \Leftrightarrow V_{in3} = 2\angle 180°\ \text{V}$$

$$\frac{-j}{\omega_3 C} = \frac{-j}{3 \times 10^3 \times 10^{-3}/3} = -j1$$

$$Z_3 = 1 - j1 = \sqrt{2}\angle -45°$$

$$I_{C3} = \frac{V_{in3}}{Z_3} = \frac{2\angle 180°}{\sqrt{2}\angle -45°} = 1.41\angle 225° = 1.41\angle(225 - 135)° \tag{7.130}$$
$$\Leftrightarrow i_{C3}(t) = 1.41\cos(3\omega t - 135°)\ \text{A}$$

From (7.129) and (7.130),

$$i_c(t) = i_{c1}(t) + i_{c3}(t) = 1.90\cos(\omega t + 71.6°) + 1.41\cos(3\omega t - 135°) \tag{7.131}$$

b. The average power delivered by the voltage source is

$$\begin{aligned} P_{ave} &= V_{1RMS} I_{1RMS}\cos\theta_1 + V_{3RMS} I_{3RMS}\cos\theta_3 \\ &= \frac{6}{\sqrt{2}} \cdot \frac{1.90}{\sqrt{2}}\cos(71.6°) + \frac{2}{\sqrt{2}} \cdot \frac{1.41}{\sqrt{2}}\cos(-135°) \end{aligned} \tag{7.132}$$

or

$$P_{ave} = 0.8\ \text{w} \tag{7.133}$$

Check:

The average power absorbed by the capacitor is zero, and therefore, the average power absorbed by the resistor, must be equal to the average power delivered by the source. The average power absorbed by the resistor is

$$P_{ave} = \frac{1}{2} I_{max}^2 R = \frac{1}{2}(I_{1max}^2 - I_{3max}^2) = \frac{1}{2}(1.90^2 - 1.41^2) = 0.8\ \text{w}$$

7.13 Evaluation of Fourier Coefficients Using Excel®

The use of Fourier series is not restricted to electric circuit analysis. It is also applied in the analysis of the behavior of physical systems subjected to periodic disturbances. Examples include cable stress analysis in suspension bridges, and mechanical vibrations.

Quite often, it is necessary to construct the Fourier expansion of a function based on observed values instead of an analytic expression. Examples are meteorological or economic quantities whose period may be a day, a week, a month or even a year. In these situations, we need to evaluate the integral(s) using numerical integration.

The procedure presented here, will work for both the waveforms that have an analytical solution and those that do not. Even though we may already know the Fourier series from analytical methods, we can use this procedure to check our results.

Consider the waveform of $f(x)$ shown in Figure 7.47, were we have divided it into small pulses of width Δx. Obviously, the more pulses we use, the better the approximation.

If the time axis is in degrees, we can choose Δx to be $2.5°$ and it is convenient to start at the zero point of the waveform. Then, using a spreadsheet, such as Microsoft Excel, we can divide the period $0°$ to $360°$ in $2.5°$ intervals, and enter these values in Column A of the spreadsheet.

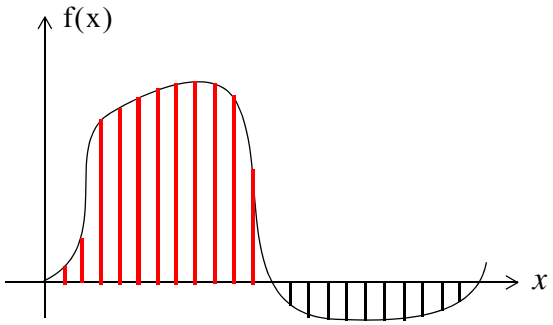


Figure 7.47. Waveform whose analytical expression is unknown

Since the arguments of the sine and the cosine are in radians, we multiply degrees by π (3.1459...) and divide by 180 to perform the conversion. We enter these in Column B and we denote them as x. In Column C we enter the corresponding values of $y = f(x)$ as measured from the waveform. In Columns D and E we enter the values of $\cos x$ and the product $y\cos x$ respectively. Similarly, we enter the values of $\sin x$ and $y\sin x$ in Columns F and G respectively.

Next, we form the sums of $y\cos x$ and $y\sin x$, we multiply these by Δx, and we divide by π to obtain the coefficients a_1 and b_1. To compute the coefficients of the higher order harmonics, we

form the products ycos2x, ysin2x, ycos3x, ysin3x, and so on, and we enter these in subsequent columns of the spreadsheet.
Figure 7.48 is a partial table showing the computation of the coefficients of the square waveform, and Figure 7.49 is a partial table showing the computation of the coefficients of a clipped sine waveform. The complete tables extend to the seventh harmonic to the right and to 360° down.

7.14 Evaluation of Fourier Coefficients Using MATLAB®

We can also use MATLAB to evaluate the coefficients of a Fourier series as illustrated with the following simple example.

Example 7.8

Let $f(t) = \cos\omega t$ where $\omega = 1$. Use the exponential Fourier series to evaluate the average value C_0 and the first 3 harmonics C_1, C_2, C_3 using MATLAB.

Solution:

We use the following MATLAB script where the statement **int(f,t,a,b)** where **f** represents the function to be integrated, **t** is a symbolic variable for the symbolic expression, and **a** and **b** are numeric values for the definite integral **a** to b.

```
syms t                                  % Define symbolic variable t
T=1;                                    % Period of the signal
w=2*pi/T;                               % radian frequency omega
for n=0:3                               % Evaluate DC component and first 3 harmonics
Cn=(1/T)*int(cos(w*t)*exp(–j*w*n*t), t, 0, 1)  % Exponential Fourier Series, relation (7.99)
end
```

MATLAB displays the following:

Cn = 0 Cn = 1/2 Cn = 0 Cn = 0

The values displayed by MATLAB indicate that $C_n = 1/2$ is the only nonzero frequency component, and since $\cos\omega t$ is an even function, all C_n coefficients in relation (7.95), Page 7–32 are real and $C_{-n} = C_n = 1/2$. Therefore,

$$f(t) = \ldots + 0 + \frac{1}{2}e^{-j\omega t} + 0 + \frac{1}{2}e^{j\omega t} + 0 + \ldots = \frac{e^{j\omega t} + e^{-j\omega t}}{2} = \cos\omega t$$

Also, for the trigonometric Fourier series, from (7.100), Page 7–33,

$$a_n = C_n + C_{-n} = \frac{1}{2} + \frac{1}{2} = 1$$

Analytical:

f(t)=4(sinwt/p+sin3wt/3p+sin5wt/5p+ ....)

Numerical:

DC=	0.000
a1=	0.000
a2=	0.000
a3=	0.000
a4=	0.000
a5=	0.000
a6=	0.000
a7=	0.000
b1=	1.273
b2=	0.000
b3=	0.424
b4=	0.000
b5=	0.254
b6=	0.000
b7=	0.180

x(deg)	x(rad)	y=f(x)	0.5*a0	cosx	ycosx	sinx	ysinx	cos2x	ycox2x	sin2x	ysin2x	cos3x	ycos3x	sin3x	ysin3x
0.0	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
2.5	0.044	1.000	0.044	0.999	0.999	0.044	0.044	0.996	0.996	0.087	0.087	0.991	0.991	0.131	0.131
5.0	0.087	1.000	0.044	0.996	0.996	0.087	0.087	0.985	0.985	0.174	0.174	0.966	0.966	0.259	0.259
7.5	0.131	1.000	0.044	0.991	0.991	0.131	0.131	0.966	0.966	0.259	0.259	0.924	0.924	0.383	0.383
10.0	0.175	1.000	0.044	0.985	0.985	0.174	0.174	0.940	0.940	0.342	0.342	0.866	0.866	0.500	0.500
12.5	0.218	1.000	0.044	0.976	0.976	0.216	0.216	0.906	0.906	0.423	0.423	0.793	0.793	0.609	0.609
15.0	0.262	1.000	0.044	0.966	0.966	0.259	0.259	0.866	0.866	0.500	0.500	0.707	0.707	0.707	0.707
17.5	0.305	1.000	0.044	0.954	0.954	0.301	0.301	0.819	0.819	0.574	0.574	0.609	0.609	0.793	0.793
20.0	0.349	1.000	0.044	0.940	0.940	0.342	0.342	0.766	0.766	0.643	0.643	0.500	0.500	0.866	0.866
22.5	0.393	1.000	0.044	0.924	0.924	0.383	0.383	0.707	0.707	0.707	0.707	0.383	0.383	0.924	0.924
25.0	0.436	1.000	0.044	0.906	0.906	0.423	0.423	0.643	0.643	0.766	0.766	0.259	0.259	0.966	0.966
27.5	0.480	1.000	0.044	0.887	0.887	0.462	0.462	0.574	0.574	0.819	0.819	0.131	0.131	0.991	0.991
30.0	0.524	1.000	0.044	0.866	0.866	0.500	0.500	0.500	0.500	0.866	0.866	0.000	0.000	1.000	1.000
32.5	0.567	1.000	0.044	0.843	0.843	0.537	0.537	0.423	0.423	0.906	0.906	-0.131	-0.131	0.991	0.991
35.0	0.611	1.000	0.044	0.819	0.819	0.574	0.574	0.342	0.342	0.940	0.940	-0.259	-0.259	0.966	0.966
37.5	0.654	1.000	0.044	0.793	0.793	0.609	0.609	0.259	0.259	0.966	0.966	-0.383	-0.383	0.924	0.924
40.0	0.698	1.000	0.044	0.766	0.766	0.643	0.643	0.174	0.174	0.985	0.985	-0.500	-0.500	0.866	0.866
42.5	0.742	1.000	0.044	0.737	0.737	0.676	0.676	0.087	0.087	0.996	0.996	-0.609	-0.609	0.793	0.793
45.0	0.785	1.000	0.044	0.707	0.707	0.707	0.707	0.000	0.000	1.000	1.000	-0.707	-0.707	0.707	0.707
47.5	0.829	1.000	0.044	0.676	0.676	0.737	0.737	-0.087	-0.087	0.996	0.996	-0.793	-0.793	0.609	0.609
50.0	0.873	1.000	0.044	0.643	0.643	0.766	0.766	-0.174	-0.174	0.985	0.985	-0.866	-0.866	0.500	0.500

Figure 7.48. Numerical computation of the coefficients of the square waveform (partial listing)

Analytical:

f(t)=unknown

Numerical:

DC=	0.000		
a1=	0.000	b1=	0.609
a2=	0.000	b2=	0.000
a3=	0.000	b3=	0.138
a4=	0.000	b4=	0.000
a5=	0.000	b5=	0.028
a6=	0.000	b6=	0.000
a7=	0.000	b7=	-0.010

x(deg)	x(rad)	y=f(x)	$0.5*a_0$	cosx	ycosx	sinx	ysinx	cos2x	ycox2x	sin2x	ysin2x	cos3x	ycos3x	sin3x	ysin3x
0.0	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
2.5	0.044	0.044	0.002	0.999	0.044	0.044	0.002	0.996	0.043	0.087	0.004	0.991	0.043	0.131	0.006
5.0	0.087	0.087	0.004	0.996	0.087	0.087	0.008	0.985	0.086	0.174	0.015	0.966	0.084	0.259	0.023
7.5	0.131	0.131	0.006	0.991	0.129	0.131	0.017	0.966	0.126	0.259	0.034	0.924	0.121	0.383	0.050
10.0	0.175	0.174	0.008	0.985	0.171	0.174	0.030	0.940	0.163	0.342	0.059	0.866	0.150	0.500	0.087
12.5	0.218	0.216	0.009	0.976	0.211	0.216	0.047	0.906	0.196	0.423	0.091	0.793	0.172	0.609	0.132
15.0	0.262	0.259	0.011	0.966	0.250	0.259	0.067	0.866	0.224	0.500	0.129	0.707	0.183	0.707	0.183
17.5	0.305	0.301	0.013	0.954	0.287	0.301	0.090	0.819	0.246	0.574	0.172	0.609	0.183	0.793	0.239
20.0	0.349	0.342	0.015	0.940	0.321	0.342	0.117	0.766	0.262	0.643	0.220	0.500	0.171	0.866	0.296
22.5	0.393	0.383	0.017	0.924	0.354	0.383	0.146	0.707	0.271	0.707	0.271	0.383	0.146	0.924	0.354
25.0	0.436	0.423	0.018	0.906	0.383	0.423	0.179	0.643	0.272	0.766	0.324	0.259	0.109	0.966	0.408
27.5	0.480	0.462	0.020	0.887	0.410	0.462	0.213	0.574	0.265	0.819	0.378	0.131	0.060	0.991	0.458
30.0	0.524	0.500	0.022	0.866	0.433	0.500	0.250	0.500	0.250	0.866	0.433	0.000	0.000	1.000	0.500
32.5	0.567	0.500	0.022	0.843	0.422	0.537	0.269	0.423	0.211	0.906	0.453	-0.131	-0.065	0.991	0.496
35.0	0.611	0.500	0.022	0.819	0.410	0.574	0.287	0.342	0.171	0.940	0.470	-0.259	-0.129	0.966	0.483
37.5	0.654	0.500	0.022	0.793	0.397	0.609	0.304	0.259	0.129	0.966	0.483	-0.383	-0.191	0.924	0.462
40.0	0.698	0.500	0.022	0.766	0.383	0.643	0.321	0.174	0.087	0.985	0.492	-0.500	-0.250	0.866	0.433
42.5	0.742	0.500	0.022	0.737	0.369	0.676	0.338	0.087	0.044	0.996	0.498	-0.609	-0.304	0.793	0.397
45.0	0.785	0.500	0.022	0.707	0.354	0.707	0.354	0.000	0.000	1.000	0.500	-0.707	-0.354	0.707	0.354
47.5	0.829	0.500	0.022	0.676	0.338	0.737	0.369	-0.087	-0.044	0.996	0.498	-0.793	-0.397	0.609	0.304
50.0	0.873	0.500	0.022	0.643	0.321	0.766	0.383	-0.174	-0.087	0.985	0.492	-0.866	-0.433	0.500	0.250

Figure 7.49. Numerical computation of the coefficients of a clipped sine waveform (partial listing)

7.15 Summary

- Any periodic waveform $f(t)$ can be expressed as

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty}(a_n \cos n\omega t + b_n \sin n\omega t)$$

where the first term $a_0/2$ is a constant, and represents the DC (average) component of $f(t)$. The terms with the coefficients a_1 and b_1 together, represent the fundamental frequency component ω. Likewise, the terms with the coefficients a_2 and b_2 together, represent the second harmonic component 2ω, and so on. The coefficients a_0, a_n, and b_n are found from the following relations:

$$\frac{1}{2}a_0 = \frac{1}{2\pi}\int_0^{2\pi} f(t)dt$$

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\cos nt\,dt$$

$$b_n = \frac{1}{\pi}\int_0^{2\pi} f(t)\sin nt\,dt$$

- If a waveform has odd symmetry, that is, if it is an odd function, the series will consist of sine terms only. We recall that odd functions are those for which $-f(-t) = f(t)$.
- If a waveform has even symmetry, that is, if it is an even function, the series will consist of cosine terms only, and a_0 may or may not be zero. We recall that even functions are those for which $f(-t) = f(t)$
- A periodic waveform with period T, has half–wave symmetry if

$$-f(t + T/2) = f(t)$$

that is, the shape of the negative half–cycle of the waveform is the same as that of the positive half-cycle, but inverted. If a waveform has half–wave symmetry only odd (odd cosine and odd sine) harmonics will be present. In other words, all even (even cosine and even sine) harmonics will be zero.

- The trigonometric Fourier series for the square waveform with odd symmetry is

$$f(t) = \frac{4A}{\pi}\left(\sin\omega t + \frac{1}{3}\sin 3\omega t + \frac{1}{5}\sin 5\omega t + \ldots\right) = \frac{4A}{\pi}\sum_{n=\text{odd}}\frac{1}{n}\sin n\omega t$$

- The trigonometric Fourier series for the square waveform with even symmetry is

$$f(t) = \frac{4A}{\pi}\left(\cos\omega t - \frac{1}{3}\cos 3\omega t + \frac{1}{5}\cos 5\omega t - \ldots\right) = \frac{4A}{\pi}\sum_{n\,=\,\text{odd}}(-1)^{\frac{(n-1)}{2}}\frac{1}{n}\cos n\omega t$$

- The trigonometric Fourier series for the sawtooth waveform with odd symmetry is

$$f(t) = \frac{2A}{\pi}\left(\sin\omega t - \frac{1}{2}\sin 2\omega t + \frac{1}{3}\sin 3\omega t - \frac{1}{4}\sin 4\omega t + \ldots\right) = \frac{2A}{\pi}\sum(-1)^{n-1}\frac{1}{n}\sin n\omega t$$

- The trigonometric Fourier series for the triangular waveform with odd symmetry is

$$f(t) = \frac{8A}{\pi^2}\left(\sin\omega t - \frac{1}{9}\sin 3\omega t + \frac{1}{25}\sin 5\omega t - \frac{1}{49}\sin 7\omega t + \ldots\right) = \frac{8A}{\pi^2}\sum_{n\,=\,\text{odd}}(-1)^{\frac{(n-1)}{2}}\frac{1}{n^2}\sin n\omega t$$

- The trigonometric Fourier series for the half–wave rectifier with no symmetry is

$$f(t) = \frac{A}{\pi} + \frac{A}{2}\sin t - \frac{A}{\pi}\left[\frac{\cos 2t}{3} + \frac{\cos 4t}{15} + \frac{\cos 6t}{35} + \frac{\cos 8t}{63} + \ldots\right]$$

- The trigonometric form of the Fourier series for the full–wave rectifier with even symmetry is

$$f(t) = \frac{2A}{\pi} - \frac{4A}{\pi}\sum_{n\,=\,2,\,4,\,6,\,\ldots}^{\infty}\frac{1}{(n^2-1)}\cos n\omega t$$

- The Fourier series are often expressed in exponential form as

$$f(t) = \ldots + C_{-2}e^{-j2\omega t} + C_{-1}e^{-j\omega t} + C_0 + C_1 e^{j\omega t} + C_2 e^{j2\omega t} + \ldots$$

  where the C_i coefficients are related to the trigonometric form coefficients as

$$C_{-n} = \frac{1}{2}\left(a_n - \frac{b_n}{j}\right) = \frac{1}{2}(a_n + jb_n)$$

$$C_n = \frac{1}{2}\left(a_n + \frac{b_n}{j}\right) = \frac{1}{2}(a_n - jb_n)$$

$$C_0 = \frac{1}{2}a_0$$

- The C_i coefficients, except C_0, are complex, and appear as complex conjugate pairs, that is,

$$C_{-n} = C_n^*$$

- In general, for $\omega \neq 1$,

$$C_n = \frac{1}{T}\int_0^T f(t)e^{-jn\omega t}d(\omega t) = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-jn\omega t}d(\omega t)$$

- We can derive the trigonometric Fourier series from the exponential series from the relations

$$a_n = C_n + C_{-n}$$

and

$$b_n = j(C_n - C_{-n})$$

- For even functions, all coefficients C_i are real
- For odd functions, all coefficients C_i are imaginary
- If there is half–wave symmetry, $C_n = 0$ for $n = \text{even}$
- $C_{-n} = C_n^*$ always
- A line spectrum is a plot that shows the amplitudes of the harmonics on a frequency scale.
- The frequency components of a recurrent rectangular pulse follow a $\sin x/x$ form.
- The line spectrum of an impulse train consists of a train of equal amplitude, and are equally spaced harmonics.
- The RMS value of a waveform consisting of sinusoids of different frequencies, is equal to the square root of the sum of the squares of the RMS values of each sinusoid. Thus,

$$I_{RMS} = \sqrt{I_0^2 + I_{1\,RMS}^2 + I_{2\,RMS}^2 + \ldots + I_{N\,RMS}^2}$$

or

$$I_{RMS} = \sqrt{I_0^2 + \frac{1}{2}I_{1\,m}^2 + \frac{1}{2}I_{2\,m}^2 + \ldots + \frac{1}{2}I_{N\,m}^2}$$

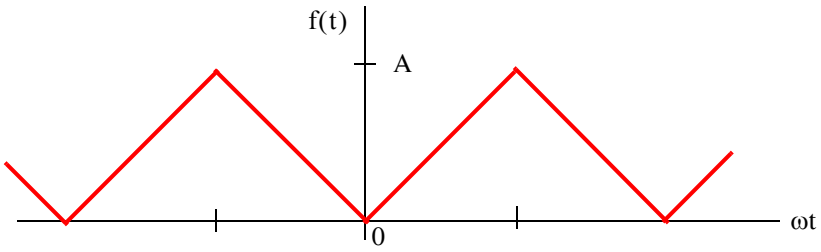
- We can compute the average power of a Fourier series from the relation

$$\begin{aligned} P_{ave} &= P_{dc} + P_{1ave} + P_{2ave} + \ldots \\ &= V_{dc}I_{dc} + V_{1RMS}I_{1RMS}\cos\theta_1 + V_{2RMS}I_{2RMS}\cos\theta_2 + \ldots \end{aligned}$$

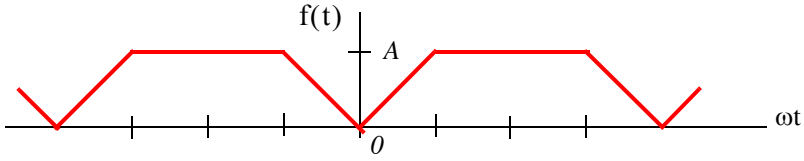
- We can evaluate the Fourier coefficients of a function based on observed values instead of an analytic expression using numerical evaluations with the aid of a spreadsheet.

7.16 Exercises

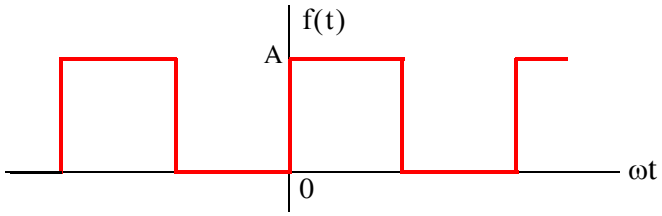
1. Compute the first 5 components of the trigonometric Fourier series for the waveform shown below. Assume $\omega = 1$.



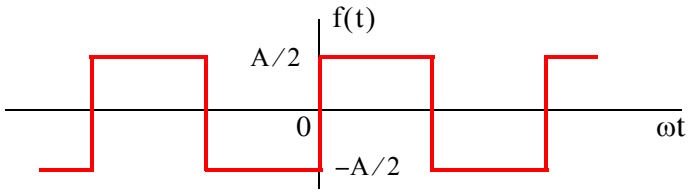
2. Compute the first 5 components of the trigonometric Fourier series for the waveform shown below. Assume $\omega = 1$.



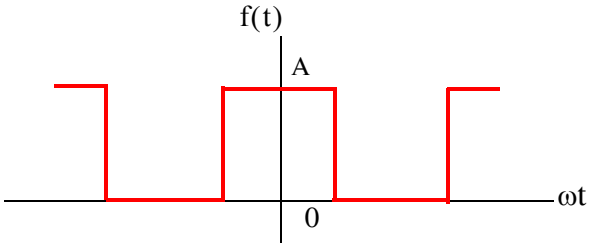
3. Compute the first 5 components of the exponential Fourier series for the waveform shown below. Assume $\omega = 1$.



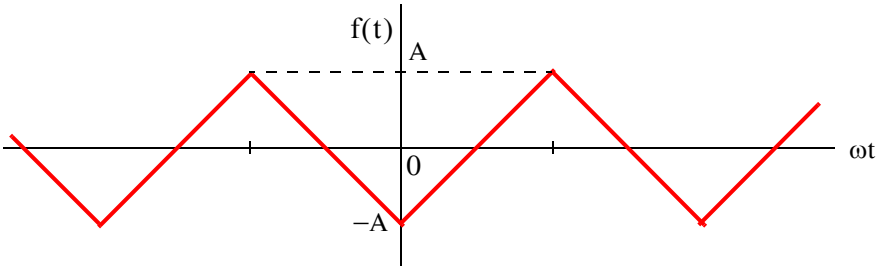
4. Compute the first 5 components of the exponential Fourier series for the waveform shown below. Assume $\omega = 1$.



5. Compute the first 5 components of the exponential Fourier series for the waveform shown below. Assume $\omega = 1$.

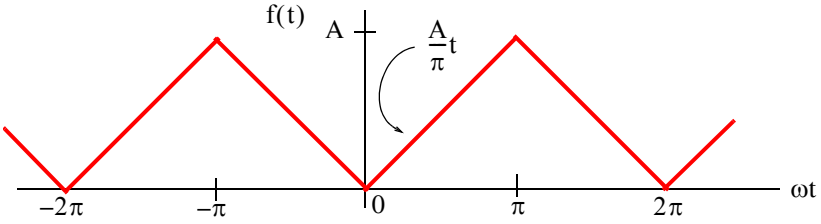


6. Compute the first 5 components of the exponential Fourier series for the waveform shown below. Assume $\omega = 1$.



7.17 Solutions to End–of–Chapter Exercises

1.



This is an even function; therefore, the series consists of cosine terms only. There is no half–wave symmetry and the average (DC component) is not zero. We will integrate from 0 to π and multiply by 2. Then,

$$a_n = \frac{2}{\pi}\int_0^{\pi}\frac{A}{\pi}t\cos nt\,dt = \frac{2A}{\pi^2}\int_0^{\pi} t\cos nt\,dt \quad (1)$$

From tables of integrals,

$$\int x\cos ax\,dx = \frac{1}{a^2}\cos ax + \frac{x}{a}\sin ax$$

and thus (1) becomes

$$a_n = \frac{2A}{\pi^2}\left(\frac{1}{n^2}\cos nt + \frac{t}{n}\sin nt\right)\Bigg|_0^{\pi} = \frac{2A}{\pi^2}\left(\frac{1}{n^2}\cos n\pi + \frac{t}{n}\sin nt\pi - \frac{1}{n^2} - 0\right)$$

and since $\sin nt\pi = 0$ for all integer n,

$$a_n = \frac{2A}{\pi^2}\left(\frac{1}{n^2}\cos n\pi - \frac{1}{n^2}\right) = \frac{2A}{n^2\pi^2}(\cos n\pi - 1) \quad (2)$$

We cannot evaluate the average $(1/2)a_0$ from (2); we must use (1). Then, for $n = 0$,

$$\frac{1}{2}a_0 = \frac{2A}{2\pi^2}\int_0^{\pi} t\,dt = \frac{A}{\pi^2}\cdot\frac{t^2}{2}\Bigg|_0^{\pi} = \frac{A}{\pi^2}\cdot\frac{\pi^2}{2}$$

or

$$(1/2)/a_0 = A/2$$

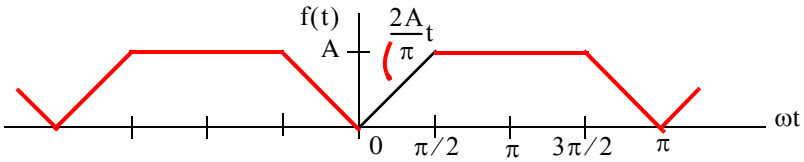
We observe from (2) that for $n = \text{even}$, $a_{n=\text{even}} = 0$. Then,

$$\text{for } n = 1,\ a_1 = -\frac{4A}{\pi^2},\ \text{for } n = 3,\ a_3 = \frac{-4A}{3^2\pi^2},\ \text{for } n = 5,\ a_5 = -\frac{4A}{5^2\pi^2},\ \text{for } n = 7,\ a_3 = \frac{-4A}{7^2\pi^2}$$

and so on. Therefore,

$$f(t) = \frac{1}{2}a_0 - \frac{4A}{\pi^2}\left(\cos t + \frac{1}{9}\cos 3t + \frac{1}{25}\cos 5t + \frac{1}{49}\cos 7t + \ldots\right) = \frac{A}{2} - \frac{4A}{\pi}\sum_{n=\text{odd}}^{\infty}\frac{1}{n^2}\cos nt$$

2.



This is an even function; therefore, the series consists of cosine terms only. There is no half–wave symmetry and the average (DC component) is not zero.

$$\text{Average} = \frac{1}{2}a_0 = \frac{\text{Area}}{\text{Period}} = \frac{2\times[(A/2)\cdot(\pi/2)] + A\pi}{2\pi} = \frac{3A\cdot(\pi/2)}{2\pi} = \frac{3A}{4}$$

$$a_n = \frac{2}{\pi}\int_0^{\pi/2}\frac{2A}{\pi}t\cos nt\,dt + \frac{2}{\pi}\int_{\pi/2}^{\pi}A\cos nt\,dt \quad (1)$$

and with

$$\int x\cos ax\,dx = \frac{1}{a^2}\cos ax + \frac{x}{a}\sin ax = \frac{1}{a^2}(\cos ax + ax\sin ax)$$

relation (1) above simplifies to

$$a_n = \frac{4A}{\pi^2}\left[\frac{1}{n^2}(\cos nt + nt\sin nt)\right]\Bigg|_0^{\pi/2} + \frac{2A}{n\pi}\sin nt\Big|_{\pi/2}^{\pi}$$

$$= \frac{4A}{n^2\pi^2}\left(\cos\frac{n\pi}{2} + \frac{n\pi}{2}\sin\frac{n\pi}{2} - 1 - 0\right) + \frac{2A}{n\pi}\left(\sin n\pi - \sin\frac{n\pi}{2}\right)$$

and since $\sin nt\pi = 0$ for all integer n,

$$a_n = \frac{4A}{n^2\pi^2}\cos\frac{n\pi}{2} + \frac{2A}{n\pi}\sin\frac{n\pi}{2} - \frac{4A}{n^2\pi^2} - \frac{2A}{n\pi}\sin\frac{n\pi}{2} = \frac{4A}{n^2\pi^2}\left(\cos\frac{n\pi}{2} - 1\right)$$

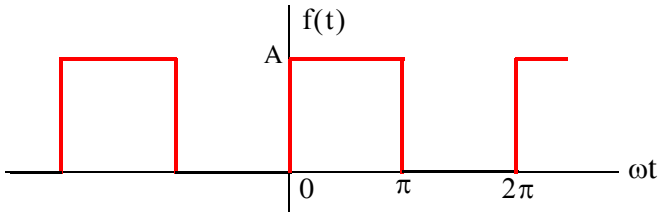
$$\text{for } n = 1,\ a_1 = \frac{4A}{\pi^2}(0-1) = -\frac{4A}{\pi^2},\ \text{for } n = 2,\ a_2 = \frac{4A}{4\pi^2}(-1-1) = -\frac{2A}{\pi^2}$$

$$\text{for } n = 3,\ a_3 = \frac{4A}{9\pi^2}(0-1) = -\frac{4A}{9\pi^2},\ \text{for } n = 4,\ a_4 = \frac{-4A}{7^2\pi^2}(1-1) = 0$$

We observe that the fourth harmonic and all its multiples are zero. Therefore,

$$f(t) = \frac{3A}{4} - \frac{4A}{\pi^2}\left(\cos t + \frac{1}{2}\cos 2t + \frac{1}{9}\cos 3t + \ldots\right)$$

3.



This is neither an even nor an odd function and has no half–wave symmetry; therefore, the series consists of both cosine and sine terms. The average (DC component) is not zero. Then,

$$C_n = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-jn\omega t}d(\omega t)$$

and with $\omega = 1$

$$C_n = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-jnt}dt = \frac{1}{2\pi}\left[\int_0^{\pi} Ae^{-jnt}dt + \int_{\pi}^{2\pi} 0e^{-jnt}dt\right] = \frac{A}{2\pi}\int_0^{\pi} e^{-jnt}dt$$

The DC value is

$$C_0 = \frac{A}{2\pi}\int_0^{\pi} e^0 dt = \frac{A}{2\pi}t\Big|_0^{\pi} = \frac{A}{2}$$

For $n \neq 0$,

$$C_n = \frac{A}{2\pi}\int_0^{\pi} e^{-jnt}dt = \frac{A}{-j2n\pi}e^{-jnt}\Big|_0^{\pi} = \frac{A}{j2n\pi}(1 - e^{-jn\pi})$$

Recalling that

$$e^{-jn\pi} = \cos n\pi - j\sin n\pi$$

for $n = \text{even}$, $e^{-jn\pi} = 1$ and for $n = \text{odd}$, $e^{-jn\pi} = -1$. Then,

$$C_{n=\text{even}} = \frac{A}{j2n\pi}(1 - 1) = 0$$

and

$$C_{n=\text{odd}} = \frac{A}{j2n\pi}[1 - (-1)] = \frac{A}{jn\pi}$$

By substitution into

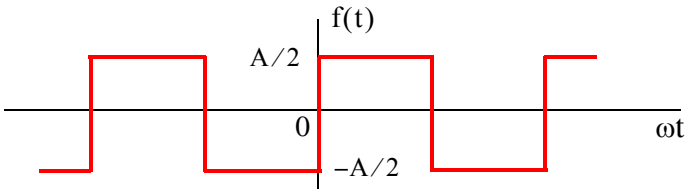
$$f(t) = \ldots + C_{-2}e^{-j2\omega t} + C_{-1}e^{-j\omega t} + C_0 + C_1 e^{j\omega t} + C_2 e^{j2\omega t} + \ldots$$

we find that

$$f(t) = \frac{A}{2} + \frac{A}{j\pi}\left(\ldots - \frac{1}{3}e^{-j3\omega t} - e^{-j\omega t} + e^{j\omega t} + \frac{1}{3}e^{j3\omega t} + \ldots\right)$$

The minus (–) sign of the first two terms within the parentheses results from the fact that $C_{-n} = C_n^*$. For instance, since $C_1 = 2A/j\pi$, it follows that $C_{-1} = C_1^* = -2A/j\pi$. We observe that $f(t)$ is complex, as expected, since there is no symmetry.

4.

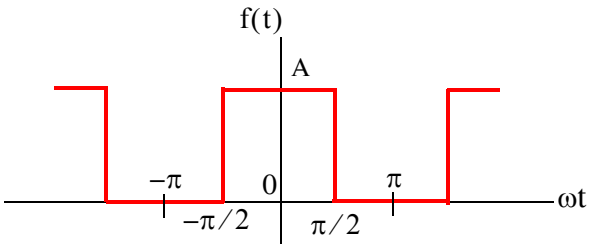


This is the same waveform as in Exercise 3 where the DC component has been removed. Then,

$$f(t) = \frac{A}{j\pi}\left(\ldots - \frac{1}{3}e^{-j3\omega t} - e^{-j\omega t} + e^{j\omega t} + \frac{1}{3}e^{j3\omega t} + \ldots\right)$$

It is also the same waveform as in Example 7.3, Page 7–34, except that the amplitude is halved. This waveform is an odd function and thus the expression for $f(t)$ is imaginary.

5.



This is the same waveform as in Exercise 3 where the vertical axis has been shifted to make the waveform an even function. Therefore, for this waveform C_n is real. Then,

$$C_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)e^{-jnt}dt = \frac{A}{2\pi}\int_{-\pi/2}^{\pi/2} e^{-jnt}dt$$

The DC value is

$$C_0 = \frac{A}{2\pi}t\Big|_{-\pi/2}^{\pi/2} = \frac{A}{2\pi}\left(\frac{\pi}{2} + \frac{\pi}{2}\right) = \frac{A}{2}$$

For $n \neq 0$,

$$C_n = \frac{A}{2\pi}\int_{-\pi/2}^{\pi/2} e^{-jnt}dt = \frac{A}{-j2n\pi}e^{-jnt}\Big|_{-\pi/2}^{\pi/2} = \frac{A}{-j2n\pi}(e^{-jn\pi/2} - e^{jn\pi/2})$$

$$= \frac{A}{j2n\pi}(e^{jn\pi/2} - e^{-jn\pi/2}) = \frac{A}{n\pi}\left(\frac{e^{jn\pi/2} - e^{-jn\pi/2}}{j2}\right) = \frac{A}{n\pi}\sin\frac{n\pi}{2}$$

and we observe that for $n = \text{even}$, $C_n = 0$

For $n = \text{odd}$, C_n alternates in plus (+) and minus (–) signs, that is,

$$C_n = \frac{A}{n\pi} \text{ if } n = 1, 5, 9, \ldots$$

$$C_n = -\frac{A}{n\pi} \text{ if } n = 3, 7, 11, \ldots$$

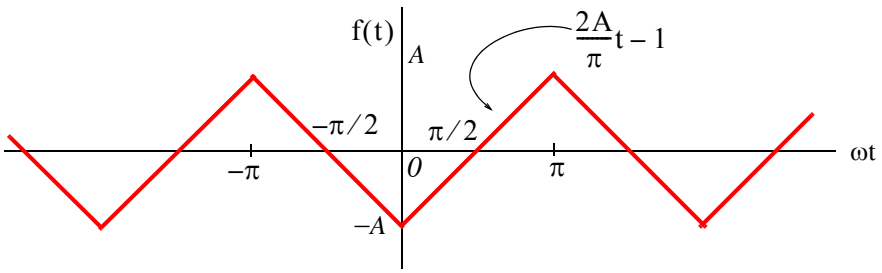
Thus,

$$f(t) = \frac{A}{2} + \sum_{n=\text{odd}} \left(\pm\frac{A}{n\pi}e^{jn\omega t}\right)$$

where the plus (+) sign is used with $n = 1, 5, 9, \ldots$ and the minus (–) sign is used with $n = 3, 7, 11, \ldots$. We can express $f(t)$ in a more compact form as

$$f(t) = \frac{A}{2} + \sum_{n=\text{odd}} (-1)^{(n-1)/2}\frac{A}{n\pi}e^{jn\omega t}$$

6.



We will find the exponential form coefficients C_n from

$$C_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)e^{-jnt}dt$$

From tables of integrals

$$\int xe^{ax}dx = \frac{e^{ax}}{a^2}(ax - 1)$$

Then,

$$C_n = \frac{1}{2\pi}\left[\int_{-\pi}^{0}\left(-\frac{2A}{\pi}t - 1\right)e^{-jnt}dt + \int_{0}^{\pi}\left(\frac{2A}{\pi}t - 1\right)e^{-jnt}dt\right]$$

Integrating and rearranging terms we obtain

$$C_n = \frac{1}{2\pi}\left[-\frac{4A}{n^2\pi} + \frac{4A}{n^2\pi}\left(n\pi\cdot\frac{e^{jn\pi} - e^{-jn\pi}}{j2} + \frac{e^{jn\pi} + e^{-jn\pi}}{2}\right) - \frac{2A}{n}\cdot\frac{e^{jn\pi} - e^{-jn\pi}}{j2}\right]$$

$$= \frac{4A}{2n^2\pi^2}\left(-1 + n\pi\sin n\pi + \cos n\pi - \frac{n\pi}{2}\sin n\pi\right)$$

and since $\sin n\pi = 0$ for all integer n,

$$C_n = \frac{2A}{n^2\pi^2}(\cos n\pi - 1)$$

For $n = \text{even}$, $C_n = 0$ and for $n = \text{odd}$, $\cos n\pi = -1$, and $C_n = \frac{-4A}{n^2\pi^2}$

Also, by inspection, the DC component $C_0 = 0$. Then,

$$f(t) = -\frac{4A}{\pi^2}\left(\ldots + \frac{1}{9}e^{-j3\omega t} + e^{-j\omega t} + e^{j\omega t} + \frac{1}{9}e^{j3\omega t} + \ldots\right)$$

The coefficients of the terms $e^{-j3\omega t}$ and $e^{-j\omega t}$ are positive because all coefficients of C_n are real. This is to be expected since $f(t)$ is an even function. It also has half–wave symmetry and thus $C_n = 0$ for $n = \text{even}$ as we've found.

Chapter 8

The Fourier Transform

This chapter introduces the Fourier Transform, also known as the Fourier Integral. The definition, theorems, and properties are presented and proved. The Fourier transforms of the most common functions are derived, the system function is defined, and several examples are provided to illustrate its application in circuit analysis.

8.1 Definition and Special Forms

We recall that the Fourier series for periodic functions of time, such as those we discussed on the previous chapter, produce discrete line spectra with non-zero values only at specific frequencies referred to as harmonics. However, other functions of interest such as the unit step, the unit impulse, the unit ramp, and a single rectangular pulse are not periodic functions. The frequency spectra for these functions are continuous as we will see later in this chapter.

We may think of a non-periodic signal as one arising from a periodic signal in which the period extents from $-\infty$ to $+\infty$. Then, for a signal that is a function of time with period from $-\infty$ to $+\infty$, we form the integral

$$\boxed{F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt} \tag{8.1}$$

and assuming that it exists for every value of the radian frequency ω, *we call the function* $F(\omega)$ *the Fourier transform or the Fourier integral.*

The Fourier transform is, in general, a complex function. We can express it as the sum of its real and imaginary components, or in exponential form, that is, as

$$F(\omega) = Re\{F(\omega)\} + jIm\{F(\omega)\} = |F(\omega)|e^{j\varphi(\omega)} \tag{8.2}$$

The *Inverse Fourier transform* is defined as

$$\boxed{f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega} \tag{8.3}$$

We will often use the following notation to express the Fourier transform and its inverse.

$$\mathcal{F}\{f(t)\} = F(\omega) \tag{8.4}$$

and

$$\mathcal{F}^{-1}\{F(\omega)\} = f(t) \tag{8.5}$$

8.2 Special Forms of the Fourier Transform

The time function $f(t)$ is, in general, complex, and thus we can express it as the sum of its real and imaginary parts, that is, as

$$f(t) = f_{Re}(t) + j f_{Im}(t) \tag{8.6}$$

The subscripts Re and Im will be used often to denote the real and imaginary parts respectively. These notations have the same meaning as $Re\{f(t)\}$ and $Im\{f(t)\}$.

By substitution of (8.6) into the Fourier integral of (8.1), we obtain

$$F(\omega) = \int_{-\infty}^{\infty} f_{Re}(t)e^{-j\omega t}dt + j\int_{-\infty}^{\infty} f_{Im}(t)e^{-j\omega t}dt \tag{8.7}$$

and by Euler's identity

$$F(\omega) = \int_{-\infty}^{\infty} [\, f_{Re}(t)\cos\omega t + f_{Im}(t)\sin\omega t]dt - j\int_{-\infty}^{\infty} [\, f_{Re}(t)\sin\omega t - f_{Im}(t)\cos\omega t]dt \tag{8.8}$$

From (8.8), we see that the real and imaginary parts of $F(\omega)$ are

$$F_{Re}(\omega) = \int_{-\infty}^{\infty} [\, f_{Re}(t)\cos\omega t + f_{Im}(t)\sin\omega t]dt \tag{8.9}$$

and

$$F_{Im}(\omega) = -\int_{-\infty}^{\infty} [\, f_{Re}(t)\sin\omega t - f_{Im}(t)\cos\omega t]dt \tag{8.10}$$

We can derive similar forms for the Inverse Fourier transform as follows:

Substitution of (8.2) into (8.3) yields

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} [F_{Re}(\omega) + jF_{Im}(\omega)]e^{j\omega t}d\omega \tag{8.11}$$

and by Euler's identity,

$$\begin{aligned} f(t) = {} & \frac{1}{2\pi}\int_{-\infty}^{\infty} [F_{Re}(\omega)\cos\omega t - F_{Im}(\omega)\sin\omega t]d\omega \\ & + j\frac{1}{2\pi}\int_{-\infty}^{\infty} [F_{Re}(\omega)\sin\omega t + F_{Im}(\omega)\cos\omega t]d\omega \end{aligned} \tag{8.12}$$

Therefore, the real and imaginary parts of $f(t)$ in terms of the Inverse Fourier transform are

$$\boxed{f_{Re}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} [F_{Re}(\omega)\cos\omega t - F_{Im}(\omega)\sin\omega t]d\omega} \tag{8.13}$$

and

$$f_{Im}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}[F_{Re}(\omega)\sin\omega t + F_{Im}(\omega)\cos\omega t]d\omega \tag{8.14}$$

Now, we will use the above relations to derive the time to frequency domain correspondence for real, imaginary, even, and odd functions in both the time and the frequency domains. We will show these in tabular form, as indicated in Table 8.1. The derivations are shown in Subsections 8.2.1 and 8.2.2.

TABLE 8.1 Time Domain and Frequency Domain Correspondence (Refer to Tables 8.2 – 8.7)

f(t)	**F(ω)**				
	Real	**Imaginary**	**Complex**	**Even**	**Odd**
Real					
Real and Even					
Real and Odd					
Imaginary					
Imaginary and Even					
Imaginary and Odd					

8.2.1 Real Time Functions

If f(t) is real, (8.9) and (8.10) reduce to

$$F_{Re}(\omega) = \int_{-\infty}^{\infty} f_{Re}(t)\cos\omega t dt \tag{8.15}$$

and

$$F_{Im}(\omega) = -\int_{-\infty}^{\infty} f_{Re}(t)\sin\omega t dt \tag{8.16}$$

Conclusion: If* f(t) *is real,* F(ω) *is, in general, complex. We indicate this result with a check mark in Table 8.2.

We know that any function f(t) can be expressed as the sum of an even and an odd function. Therefore, we will also consider the cases when f(t) is real and even, and when f(t) is real and odd*.

* *In our subsequent discussion, we will make use of the fact that the cosine is an even function, while the sine is an odd function. Also, the product of two odd functions or the product of two even functions will result in an even function, whereas the product of an odd function and an even function will result in an odd function.*

TABLE 8.2 Time Domain and Frequency Domain Correspondence (Refer also to Tables 8.3 – 8.7)

f(t)	F(ω)				
	Real	Imaginary	Complex	Even	Odd
Real			✔		
Real and Even					
Real and Odd					
Imaginary					
Imaginary and Even					
Imaginary and Odd					

a. $f_{Re}(t)$ is even

If $f_{Re}(-t) = f_{Re}(t)$, the product $f_{Re}(t)\cos\omega t$, with respect to t, is even, while the product $f_{Re}(t)\sin\omega t$ is odd. In this case, (8.15) and (8.16) reduce to:

$$\boxed{F_{Re}(\omega) = 2\int_0^\infty f_{Re}(t)\cos\omega t dt \qquad f_{Re}(t) = \text{even}} \tag{8.17}$$

and

$$\boxed{F_{Im}(\omega) = -\int_{-\infty}^\infty f_{Re}(t)\sin\omega t dt = 0 \qquad f_{Re}(t) = \text{even}} \tag{8.18}$$

Therefore, if $f_{Re}(t) = \text{even}$, $F(\omega)$ is real as seen in (8.17).

To determine whether $F(\omega)$ is even or odd when $f_{Re}(t) = \text{even}$, we must perform a test for evenness or oddness with respect to ω. Thus, substitution of $-\omega$ for ω in (8.17), yields

$$F_{Re}(-\omega) = 2\int_0^\infty f_{Re}(t)\cos(-\omega)t dt = 2\int_0^\infty f_{Re}(t)\cos\omega t dt = F_{Re}(\omega) \tag{8.19}$$

Conclusion: If $f(t)$ is real and even, $F(\omega)$ is also real and even. We indicate this result in Table 8.3.

b. $f_{Re}(t)$ is odd

If $-f_{Re}(-t) = f_{Re}(t)$, the product $f_{Re}(t)\cos\omega t$, with respect to t, is odd, while the product $f_{Re}(t)\sin\omega t$ is even. In this case, (8.15) and (8.16) reduce to:

TABLE 8.3 Time Domain and Frequency Domain Correspondence (Refer also to Tables 8.4 – 8.7)

f(t)	F(ω)				
	Real	Imaginary	Complex	Even	Odd
Real			✔		
Real and Even	✔			✔	
Real and Odd					
Imaginary					
Imaginary and Even					
Imaginary and Odd					

$$F_{Re}(\omega) = \int_{-\infty}^{\infty} f_{Re}(t)\cos\omega t dt = 0 \qquad f_{Re}(t) = \text{odd} \tag{8.20}$$

and

$$F_{Im}(\omega) = -2\int_{0}^{\infty} f_{Re}(t)\sin\omega t dt \qquad f_{Re}(t) = \text{odd} \tag{8.21}$$

Therefore, if $f_{Re}(t) = \text{odd}$, $F(\omega)$ is imaginary.

To determine whether $F(\omega)$ is even or odd when $f_{Re}(t) = \text{odd}$, we perform a test for evenness or oddness with respect to ω. Thus, substitution of $-\omega$ for ω in (8.21), yields

$$F_{Im}(-\omega) = -2\int_{0}^{\infty} f_{Re}(t)\sin(-\omega)t dt = 2\int_{0}^{\infty} f_{Re}(t)\sin\omega t dt = -F_{Im}(\omega) \tag{8.22}$$

Conclusion: If f(t) is real and odd, F(ω) is imaginary and odd. We indicate this result in Table 8.4.

TABLE 8.4 Time Domain and Frequency Domain Correspondence (Refer also to Tables 8.5 – 8.7)

f(t)	F(ω)				
	Real	Imaginary	Complex	Even	Odd
Real			✔		
Real and Even	✔			✔	
Real and Odd		✔			✔
Imaginary					
Imaginary and Even					
Imaginary and Odd					

8.2.2 Imaginary Time Functions

If f(t) is imaginary, (8.9) and (8.10) reduce to

$$F_{Re}(\omega) = \int_{-\infty}^{\infty} f_{Im}(t)\sin\omega t dt \tag{8.23}$$

and

$$F_{Im}(\omega) = \int_{-\infty}^{\infty} f_{Im}(t)\cos\omega t dt \tag{8.24}$$

Conclusion: If $\mathbf{f(t)}$ ***is imaginary,*** $\mathbf{F(\omega)}$ ***is, in general, complex.*** We indicate this result in Table 8.5.

TABLE 8.5 Time Domain and Frequency Domain Correspondence (Refer also to Tables 8.6 – 8.7)

f(t)	**F(ω)**				
	Real	**Imaginary**	**Complex**	**Even**	**Odd**
Real			✔		
Real and Even	✔			✔	
Real and Odd		✔			✔
Imaginary			✔		
Imaginary and Even					
Imaginary and Odd					

Next, we will consider the cases where f(t) is imaginary and even, and f(t) is imaginary and odd.

a. $\mathbf{f_{Im}(t)}$ **is even**

If $f_{Im}(-t) = f_{Im}(t)$, the product $f_{Im}(t)\cos\omega t$, with respect to t, is even while the product $f_{Im}(t)\sin\omega t$ is odd. In this case, (8.23) and (8.24) reduce to:

$$F_{Re}(\omega) = \int_{-\infty}^{\infty} f_{Im}(t)\sin\omega t dt = 0 \qquad f_{Im}(t) = \text{even} \tag{8.25}$$

and

$$F_{Im}(\omega) = 2\int_{0}^{\infty} f_{Im}(t)\cos\omega t dt \qquad f_{Im}(t) = \text{even} \tag{8.26}$$

Therefore, if $f_{Im}(t) = \text{even}$, $F(\omega)$ is imaginary.

To determine whether $F(\omega)$ is even or odd when $f_{Im}(t) = \text{even}$, we perform a test for evenness or oddness with respect to ω. Thus, substitution of $-\omega$ for ω in (8.26) yields

$$F_{Im}(-\omega) = 2\int_0^{\infty} f_{Im}(t)\cos(-\omega)tdt = 2\int_0^{\infty} f_{Im}(t)\cos\omega tdt = F_{Im}(\omega) \tag{8.27}$$

Conclusion: If $\mathbf{f(t)}$ ***is imaginary and even,*** $\mathbf{F(\omega)}$ ***is also imaginary and even.*** We indicate this result in Table 8.6.

TABLE 8.6 Time Domain and Frequency Domain Correspondence (Refer also to Table 8.7)

f(t)	**F(ω)**				
	Real	**Imaginary**	**Complex**	**Even**	**Odd**
Real			✔		
Real and Even	✔			✔	
Real and Odd		✔			✔
Imaginary			✔		
Imaginary and Even		✔		✔	
Imaginary and Odd					

b. $\mathbf{f_{Im}(t)}$ **is odd**

If $-f_{Im}(-t) = f_{Im}(t)$, the product $f_{Im}(t)\cos\omega t$, with respect to t, is odd, while the product $f_{Im}(t)\sin\omega t$ is even. In this case, (8.23) and (8.24) reduce to

$$F_{Re}(\omega) = \int_{-\infty}^{\infty} f_{Im}(t)\sin\omega tdt = 2\int_0^{\infty} f_{Im}(t)\sin\omega tdt \qquad f_{Im}(t) = \text{odd} \tag{8.28}$$

and

$$F_{Im}(\omega) = \int_{-\infty}^{\infty} f_{Im}(t)\cos\omega tdt = 0 \qquad f_{Im}(t) = \text{odd} \tag{8.29}$$

Therefore, if $f_{Im}(t) = \text{odd}$, $F(\omega)$ is real.

To determine whether $F(\omega)$ is even or odd when $f_{Im}(t) = \text{odd}$, we perform a test for evenness or oddness with respect to ω. Thus, substitution of $-\omega$ for ω in (8.28) yields

$$F_{Re}(-\omega) = 2\int_0^{\infty} f_{Im}(t)\sin(-\omega)tdt = -2\int_0^{\infty} f_{Im}(t)\sin\omega tdt = -F_{Re}(\omega) \tag{8.30}$$

Conclusion: If $\mathbf{f(t)}$ ***is imaginary and odd,*** $\mathbf{F(\omega)}$ ***is real and odd.*** We indicate this result in Table 8.7.

TABLE 8.7 Time Domain and Frequency Domain Correspondence (Completed Table)

f(t)	**F(ω)**				
	Real	**Imaginary**	**Complex**	**Even**	**Odd**
Real			✔		
Real and Even	✔			✔	
Real and Odd		✔			✔
Imaginary			✔		
Imaginary and Even		✔		✔	
Imaginary and Odd	✔				✔

Table 8.7 is now complete and shows that if $f(t)$ is real (even or odd), the real part of $F(\omega)$ is even, and the imaginary part is odd. Then,

$$F_{Re}(-\omega) = F_{Re}(\omega) \qquad f(t) = \text{Real} \tag{8.31}$$

and

$$F_{Im}(-\omega) = -F_{Im}(\omega) \qquad f(t) = \text{Real} \tag{8.32}$$

Since,

$$F(\omega) = F_{Re}(\omega) + jF_{Im}(\omega) \tag{8.33}$$

it follows that

$$F(-\omega) = F_{Re}(-\omega) + jF_{Im}(-\omega) = F_{Re}(\omega) - jF_{Im}(\omega)$$

or

$$\boxed{F(-\omega) = F^*(\omega) \qquad f(t) = \text{Real}} \tag{8.34}$$

Now, if $F(\omega)$ of some function of time $f(t)$ is known, and $F(\omega)$ is such that $F(-\omega) = F^*(\omega)$, can we conclude that $f(t)$ is real? The answer is yes; we can verify this with (8.14), Page 8–3, which is repeated here for convenience.

$$f_{Im}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}[F_{Re}(\omega)\sin\omega t + F_{Im}(\omega)\cos\omega t]d\omega \tag{8.35}$$

We observe that the integrand of (8.35) is zero since it is an odd function with respect to ω because both products inside the brackets are odd functions*.

Therefore, $f_{Im}(t) = 0$, that is, $f(t)$ is real.

* *In relations (8.31) and (8.32) above, we determined that* $F_{Re}(\omega)$ *is even and* $F_{Im}(\omega)$ *is odd.*

Accordingly, we can state that *a necessary and sufficient condition for* $f(t)$ *to be real, is that* $F(-\omega) = F^*(\omega)$.

Also, if it is known that $f(t)$ is real, the Inverse Fourier transform of (8.3) can be simplified as follows:

From (8.13), Page 8–2,

$$f_{Re}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}[F_{Re}(\omega)\cos\omega t - F_{Im}(\omega)\sin\omega t]d\omega \tag{8.36}$$

and since the integrand is an even function with respect to ω, we rewrite (8.36) as

$$\begin{aligned} f_{Re}(t) &= 2\frac{1}{2\pi}\int_{0}^{\infty}[F_{Re}(\omega)\cos\omega t - F_{Im}(\omega)\sin\omega t]d\omega \\ &= \frac{1}{\pi}\int_{0}^{\infty}A(\omega)\cos[\omega t + \varphi(\omega)]d\omega = \frac{1}{\pi}\mathrm{Re}\int_{0}^{\infty}F(\omega)e^{j[\omega t + \varphi(\omega)]}d\omega \end{aligned} \tag{8.37}$$

8.3 Properties and Theorems of the Fourier Transform

The most common properties and theorems of the Fourier transform are described in Subsections 8.3.1 through 8.3.14 below.

8.3.1 Linearity

If $F_1(\omega)$ is the Fourier transform of $f_1(t)$, $F_2(\omega)$ is the transform of $f_2(t)$, and so on, the *linearity property of the Fourier transform* states that

$$\boxed{a_1 f_1(t) + a_2 f_2(t) + \ldots + a_n f_n(t) \Leftrightarrow a_1 F_1(\omega) + a_2 F_2(\omega) + \ldots + a_n F_n(\omega)} \tag{8.38}$$

where a_i is some arbitrary real constant.

Proof:

The proof is easily obtained from (8.1), Page 8–1, that is, the definition of the Fourier transform. The procedure is the same as for the linearity property of the Laplace transform in Chapter 2.

8.3.2 Symmetry

If $F(\omega)$ is the Fourier transform of $f(t)$, the *symmetry property of the Fourier transform* states that

$$\boxed{F(t) \Leftrightarrow 2\pi f(-\omega)} \tag{8.39}$$

that is, if in $F(\omega)$, we replace ω with t, we obtain the Fourier transform pair of (8.39).

Proof:

Since

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega$$

then,

$$2\pi f(-t) = \int_{-\infty}^{\infty} F(\omega)e^{-j\omega t}d\omega$$

Interchanging t and ω, we obtain

$$2\pi f(-\omega) = \int_{-\infty}^{\infty} F(t)e^{-j\omega t}dt$$

and (8.39) follows.

8.3.3 Time Scaling

If a is a real constant, and $F(\omega)$ is the Fourier transform of $f(t)$, then,

$$\boxed{f(at) \Leftrightarrow \frac{1}{|a|}F\left(\frac{\omega}{a}\right)} \tag{8.40}$$

that is, the *time scaling property of the Fourier transform* states that if we replace the variable t in the time domain by at, we must replace the variable ω in the frequency domain by ω/a, and divide $F(\omega/a)$ by the absolute value of a.

Proof:

We must consider both cases $a > 0$ and $a < 0$.

For $a > 0$,

$$\mathcal{F}\{f(at)\} = \int_{-\infty}^{\infty} f(at)e^{-j\omega t}dt \tag{8.41}$$

We let $at = \tau$; then, $t = \tau/a$, and (8.41) becomes

$$\mathcal{F}\{f(\tau)\} = \int_{-\infty}^{\infty} f(\tau)e^{-j\omega\left(\frac{\tau}{a}\right)}d\left(\frac{\tau}{a}\right) = \frac{1}{a}\int_{-\infty}^{\infty} f(\tau)e^{-j\left(\frac{\omega}{a}\right)\tau}d\tau = \frac{1}{a}F\left(\frac{\omega}{a}\right)$$

For $a < 0$,

$$\mathcal{F}\{f(-at)\} = \int_{-\infty}^{\infty} f(-at)e^{-j\omega t}dt$$

and making the above substitutions, we find that the multiplying factor is $-1/a$. Therefore, for $1/|a|$ we obtain (8.40).

8.3.4 Time Shifting

If $F(\omega)$ is the Fourier transform of $f(t)$, then,

$$\boxed{f(t-t_0) \Leftrightarrow F(\omega)e^{-j\omega t_0}} \tag{8.42}$$

that is, the *time shifting property of the Fourier transform* states that if we shift the time function $f(t)$ by a constant t_0, the Fourier transform magnitude does not change, but the term ωt_0 is added to its phase angle.

Proof:

$$\mathcal{F}\{f(t-t_0)\} = \int_{-\infty}^{\infty} f(t-t_0)e^{-j\omega t}dt$$

We let $t-t_0 = \tau$; then, $t = \tau + t_0$, $dt = d\tau$, and thus

$$\mathcal{F}\{f(t-t_0)\} = \int_{-\infty}^{\infty} f(\tau)e^{-j\omega(\tau+t_0)}d\tau = e^{-j\omega t_0}\int_{-\infty}^{\infty} f(\tau)e^{-j\omega(\tau)}d\tau$$

or

$$\mathcal{F}\{f(t-t_0)\} = e^{-j\omega t_0}F(\omega)$$

8.3.5 Frequency Shifting

If $F(\omega)$ is the Fourier transform of $f(t)$, then,

$$\boxed{e^{j\omega_0 t}f(t) \Leftrightarrow F(\omega-\omega_0)} \tag{8.43}$$

that is, *multiplication of the time function* $f(t)$ by $e^{j\omega_0 t}$, where ω_0 is a constant, results in shifting the Fourier transform by ω_0.

Proof:

$$\mathcal{F}\{e^{j\omega_0 t}f(t)\} = \int_{-\infty}^{\infty} e^{j\omega_0 t}f(t)e^{-j\omega t}dt$$

or

$$\mathcal{F}\{e^{j\omega_0 t}f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-j(\omega-\omega_0)}dt = F(\omega-\omega_0)$$

Also, from (8.40) and (8.43)

$$\boxed{e^{j\omega_0 t}f(at) \Leftrightarrow \frac{1}{|a|}F\left(\frac{\omega-\omega_0}{a}\right)} \tag{8.44}$$

The Frequency Shifting Property, that is, (8.43) is also used to derive the Fourier transform of the *modulated signals* $f(t)\cos\omega t$ and $f(t)\sin\omega t$. Thus, from

$$e^{j\omega_0 t}f(t) \Leftrightarrow F(\omega-\omega_0)$$

and

$$\cos\omega_0 t = \frac{e^{j\omega_0 t}+e^{-j\omega_0 t}}{2}$$

we obtain

$$\boxed{f(t)\cos\omega_0 t \Leftrightarrow \frac{F(\omega-\omega_0)+F(\omega+\omega_0)}{2}} \tag{8.45}$$

Similarly,

$$\boxed{f(t)\sin\omega_0 t \Leftrightarrow \frac{F(\omega-\omega_0)-F(\omega+\omega_0)}{j2}} \tag{8.46}$$

8.3.6 Time Differentiation

If $F(\omega)$ is the Fourier transform of $f(t)$, then,

$$\boxed{\frac{d^n}{dt^n}f(t) \Leftrightarrow (j\omega)^n F(\omega)} \tag{8.47}$$

that is, the Fourier transform of $\frac{d^n}{dt^n}f(t)$, if it exists, is $(j\omega)^n F(\omega)$.

Proof:

Differentiating the Inverse Fourier transform, we obtain

$$\frac{d^n}{dt^n}f(t) = \frac{d^n}{dt^n}\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}F(\omega)e^{j\omega t}d\omega\right) = \frac{1}{2\pi}\int_{-\infty}^{\infty}F(\omega)\frac{d^n}{dt^n}e^{j\omega t}d\omega$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty}F(\omega)(j\omega)^n e^{j\omega t}d\omega = (j\omega)^n\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}F(\omega)e^{j\omega t}d\omega\right)$$

and (8.47) follows.

8.3.7 Frequency Differentiation

If $F(\omega)$ is the Fourier transform of $f(t)$, then,

$$\boxed{(-jt)^n f(t) \Leftrightarrow \frac{d^n}{d\omega^n} F(\omega)} \tag{8.48}$$

Proof:

Using the Fourier transform definition, we obtain

$$\frac{d^n}{d\omega^n} F(\omega) = \frac{d^n}{d\omega^n}\left(\int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt\right) = \int_{-\infty}^{\infty} f(t)\frac{d^n}{d\omega^n}e^{-j\omega t}dt$$

$$= \int_{-\infty}^{\infty} f(t)(-jt)^n e^{-j\omega t}dt = (-jt)^n\int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$$

and (8.48) follows.

8.3.8 Time Integration

If $F(\omega)$ is the Fourier transform of $f(t)$, then,

$$\boxed{\int_{-\infty}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(\omega)}{j\omega} + \pi F(0)\delta(\omega)} \tag{8.49}$$

Proof:

We postpone the proof of this property until we derive the Fourier transform of the Unit Step Function $u_0(t)$ in Subsection 8.4.6, Page 8–22, and the proof for the Time Integration Property, Page 8–23. In the special case where in (8.49) $F(0) = 0$, then,

$$\int_{-\infty}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(\omega)}{j\omega} \tag{8.50}$$

and this is easily proved by integrating both sides of the Inverse Fourier transform.

8.3.9 Conjugate Time and Frequency Functions

If $F(\omega)$ is the Fourier transform of the complex function $f(t)$, then,

$$\boxed{f^*(t) \Leftrightarrow F^*(-\omega)} \tag{8.51}$$

that is, if the Fourier transform of $f(t) = f_{Re}(t) + f_{Im}(t)$ is $F(\omega)$, then, the Fourier transform of $f^*(t) = f_{Re}(t) - f_{Im}(t)$ is $F^*(-\omega)$.

Proof:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt = \int_{-\infty}^{\infty} [f_{Re}(t) + jf_{Im}(t)]e^{-j\omega t}dt$$

$$= \int_{-\infty}^{\infty} f_{Re}(t)e^{-j\omega t}dt + j\int_{-\infty}^{\infty} f_{Im}(t)e^{-j\omega t}dt$$

Then,

$$F^*(\omega) = \int_{-\infty}^{\infty} f_{Re}(t)e^{j\omega t}dt - j\int_{-\infty}^{\infty} f_{Im}(t)e^{j\omega t}dt$$

Replacing ω with $-\omega$, we obtain

$$F^*(-\omega) = \int_{-\infty}^{\infty} [f_{Re}(t) - jf_{Im}(t)]e^{-j\omega t}dt$$

and (8.51) follows.

8.3.10 Time Convolution

If $F_1(\omega)$ is the Fourier transform of $f_1(t)$, and $F_2(\omega)$ is the Fourier transform of $f_2(t)$, then,

$$\boxed{f_1(t)*f_2(t) \Leftrightarrow F_1(\omega)F_2(\omega)} \tag{8.52}$$

that is, convolution in the time domain, corresponds to multiplication in the frequency domain.

Proof:

$$\begin{aligned}\mathcal{F}\{f_1(t)*f_2(t)\} &= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)d\tau\right]e^{-j\omega t}dt \\ &= \int_{-\infty}^{\infty} f_1(\tau)\left[\int_{-\infty}^{\infty} f_2(t-\tau)e^{-j\omega t}dt\right]d\tau\end{aligned} \tag{8.53}$$

and letting $t-\tau = \sigma$, then, $dt = d\sigma$, and by substitution into (8.53),

$$\mathcal{F}\{f_1(t)*f_2(t)\} = \int_{-\infty}^{\infty} f_1(\tau)\left[\int_{-\infty}^{\infty} f_2(\sigma)e^{-j\omega\tau}e^{-j\omega\sigma}d\sigma\right]d\tau = \int_{-\infty}^{\infty} f_1(\tau)e^{-j\omega\tau}d\tau\int_{-\infty}^{\infty} f_2(\sigma)e^{-j\omega\sigma}d\sigma$$

The first integral above is $F_1(\omega)$ while the second is $F_2(\omega)$, and thus (8.52) follows.

Alternate Proof:

We can apply the time shifting property $f(t-t_0) \Leftrightarrow F(\omega)e^{-j\omega t_0}$ into the bracketed integral of (8.53); then, replacing it with $F_2(\omega)e^{-j\omega t_0}$, we obtain

$$\mathcal{F}\{f_1(t)*f_2(t)\} = \int_{-\infty}^{\infty} f_1(\tau)\left[\int_{-\infty}^{\infty} f_2(t-\tau)e^{-j\omega t}dt\right]d\tau = \int_{-\infty}^{\infty} f_1(\tau)d\tau F_2(\omega)e^{-j\omega t_0}$$

$$= \int_{-\infty}^{\infty} f_1(\tau)e^{-j\omega t}d\tau F_2(\omega) = F_1(\omega)F_2(\omega)$$

8.3.11 Frequency Convolution

If $F_1(\omega)$ is the Fourier transform of $f_1(t)$, and $F_2(\omega)$ is the Fourier transform of $f_2(t)$, then,

$$\boxed{f_1(t)f_2(t) \Leftrightarrow \frac{1}{2\pi}F_1(\omega)*F_2(\omega)} \tag{8.54}$$

that is, multiplication in the time domain, corresponds to convolution in the frequency domain divided by the constant $1/2\pi$.

Proof:

$$\mathcal{F}\{f_1(t)f_2(t)\} = \int_{-\infty}^{\infty}[f_1(t)f_2(t)]e^{-j\omega t}dt = \int_{-\infty}^{\infty}\frac{1}{2\pi}\left[\int_{-\infty}^{\infty}F_1(\chi)e^{j\chi t}d\chi\right]f_2(t)e^{-j\omega t}dt$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty}F_1(\chi)\left[\int_{-\infty}^{\infty}f_2(t)e^{-j(\omega-\chi)t}dt\right]d\chi = \frac{1}{2\pi}\int_{-\infty}^{\infty}F_1(\chi)F_2(\omega-\chi)d\chi$$

and (8.54) follows.

8.3.12 Area Under f(t)

If $F(\omega)$ is the Fourier transform of $f(t)$, then,

$$\boxed{F(0) = \int_{-\infty}^{\infty} f(t)dt} \tag{8.55}$$

that is, the area under a time function $f(t)$ is equal to the value of its Fourier transform evaluated at $\omega = 0$.

Proof:

Using the definition of $F(\omega)$ and that $e^{-j\omega t}\big|_{\omega=0} = 1$, we observe that (8.55) follows.

8.3.13 Area Under F(ω)

If $F(\omega)$ is the Fourier transform of $f(t)$, then,

$$\boxed{f(0) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)d\omega} \tag{8.56}$$

that is, the value of the time function $f(t)$, evaluated at $t = 0$, is equal to the area under its Fourier transform $F(\omega)$ times $1/2\pi$.

Proof:

In the Inverse Fourier transform of (8.3), we let $e^{j\omega t}\big|_{t=0} = 1$, and (8.56) follows.

8.3.14 Parseval's Theorem

If $F(\omega)$ is the Fourier transform of $f(t)$, *Parseval's theorem* states that

$$\boxed{\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |F(\omega)|^2 d\omega} \tag{8.57}$$

that is, if the time function $f(t)$ represents the voltage across, or the current through an $1\ \Omega$ resistor, the instantaneous power absorbed by this resistor is either v^2/R, $v^2/1$, v^2, or i^2R, i^2. Then, the integral of the magnitude squared, represents the energy (in watt-seconds or joules) dissipated by the resistor. For this reason, the integral is called *the energy of the signal*. Relation (8.57) then, states that if we do not know the energy of a time function $f(t)$, but we know the Fourier transform of this function, we can compute the energy without the need to evaluate the Inverse Fourier transform.

Proof:

From the frequency convolution property,

$$f_1(t)f_2(t) \Leftrightarrow \frac{1}{2\pi}F_1(\omega)*F_2(\omega)$$

or

$$\mathcal{F}\{f_1(t)f_2(t)\} = \int_{-\infty}^{\infty}[f_1(t)f_2(t)]e^{-j\omega t}dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} F_1(\chi)F_2(\omega-\chi)d\chi \tag{8.58}$$

Since (8.58) must hold for all values of ω, it must also be true for $\omega = 0$, and under this condition, it reduces to

$$\int_{-\infty}^{\infty}[f_1(t)f_2(t)]dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} F_1(\chi)F_2(-\chi)d\chi \tag{8.59}$$

For the special case where $f_2(t) = f_1{}^*(t)$, and the conjugate functions property $f^*(t) \Leftrightarrow F^*(-\omega)$, by substitution into (8.59), we obtain:

$$\int_{-\infty}^{\infty}[f(t)f^*(t)]dt = \frac{1}{2\pi}\int_{-\infty}^{\infty}F(\omega)F^*[-(-\omega)]d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty}F(\omega)F^*(\omega)d\omega$$

Since $f(t)f^*(t) = |f(t)|^2$ and $F(\omega)F^*(\omega) = |F(\omega)|^2$, Parseval's theorem is proved.

For convenience, the Fourier transform properties and theorems are summarized in Table 8.8.

TABLE 8.8 Fourier Transform Properties and Theorems

Property	f(t)	F(ω)
Linearity	$a_1 f_1(t) + a_2 f_2(t) + \ldots$	$a_1 F_1(\omega) + a_2 F_2(\omega) + \ldots$
Symmetry	$F(t)$	$2\pi f(-\omega)$
Time Scaling	$f(at)$	$\frac{1}{\lvert a\rvert}F\left(\frac{\omega}{a}\right)$
Time Shifting	$f(t-t_0)$	$F(\omega)e^{-j\omega t_0}$
Frequency Shifting	$e^{j\omega_0 t}f(t)$	$F(\omega-\omega_0)$
Time Differentiation	$\frac{d^n}{dt^n}f(t)$	$(j\omega)^n F(\omega)$
Frequency Differentiation	$(-jt)^n f(t)$	$\frac{d^n}{d\omega^n}F(\omega)$
Time Integration	$\int_{-\infty}^{t} f(\tau)d\tau$	$\frac{F(\omega)}{j\omega} + \pi F(0)\delta(\omega)$
Conjugate Functions	$f^*(t)$	$F^*(-\omega)$
Time Convolution	$f_1(t)*f_2(t)$	$F_1(\omega)\cdot F_2(\omega)$
Frequency Convolution	$f_1(t)\cdot f_2(t)$	$\frac{1}{2\pi}F_1(\omega)*F_2(\omega)$
Area under f(t)	$F(0) = \int_{-\infty}^{\infty} f(t)dt$	
Area under F(ω)	$f(0) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)d\omega$	
Parseval's Theorem	$\int_{-\infty}^{\infty}\lvert f(t)\rvert^2 dt = \frac{1}{2\pi}\int_{-\infty}^{\infty}\lvert F(\omega)\rvert^2 d\omega$	

8.4 Fourier Transform Pairs of Common Functions

The Fourier transform pair of the most common functions described in Subsections 8.4.1 through 8.4.9 below.

8.4.1 The Delta Function Pair

$$\boxed{\delta(t) \Leftrightarrow 1} \tag{8.60}$$

Proof:

The sifting theorem of the delta function states that

$$\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt = f(t_0)$$

and if $f(t)$ is defined at $t = 0$, then,

$$\int_{-\infty}^{\infty} f(t)\delta(t)dt = f(0)$$

By the definition of the Fourier transform

$$F(\omega) = \int_{-\infty}^{\infty} \delta(t)e^{-j\omega t}dt = e^{-j\omega t}\Big|_{t=0} = 1$$

and (8.60) follows.

Likewise, the Fourier transform for the shifted delta function $\delta(t-t_0)$ is

$$\boxed{\delta(t-t_0) \Leftrightarrow e^{-j\omega t_0}} \tag{8.61}$$

We will use the notation $f(t) \leftrightarrow F(\omega)$ to show the time domain to frequency domain correspondence. Thus, (8.60) may also be denoted as in Figure 8.1.

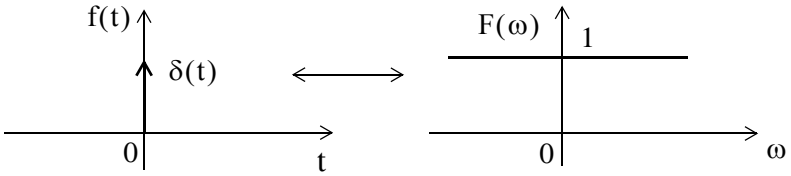


Figure 8.1. The Fourier transform of the delta function

8.4.2 The Constant Function Pair

$$\boxed{A \Leftrightarrow 2A\pi\delta(\omega)} \tag{8.62}$$

Proof:

$$\mathcal{F}^{-1}\{2A\pi\delta(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} 2A\pi\delta(\omega)e^{j\omega t}d\omega = A\int_{-\infty}^{\infty}\delta(\omega)e^{j\omega t}d\omega = Ae^{j\omega t}\Big|_{\omega=0} = A$$

and (8.62) follows.

The $f(t) \leftrightarrow F(\omega)$ correspondence is also shown in Figure 8.2.

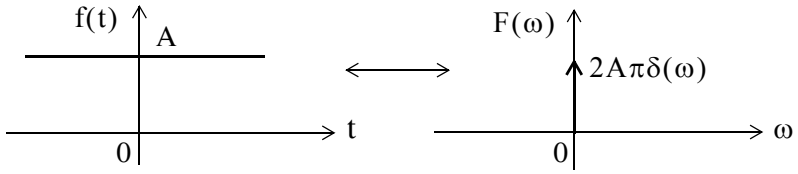


Figure 8.2. The Fourier transform of constant A

Also, by direct application of the Inverse Fourier transform, or the frequency shifting property and (8.62), we derive the transform

$$\boxed{e^{j\omega_0 t} \Leftrightarrow 2\pi\delta(\omega-\omega_0)} \tag{8.63}$$

The transform pairs of (8.62) and (8.63) can also be derived from (8.60) and (8.61) by using the symmetry property $F(t) \Leftrightarrow 2\pi f(-\omega)$

8.4.3 The Cosine Function Pair

$$\boxed{\cos\omega_0 t = \frac{1}{2}(e^{j\omega_0 t} + e^{-j\omega_0 t}) \Leftrightarrow \pi\delta(\omega-\omega_0) + \pi\delta(\omega+\omega_0)} \tag{8.64}$$

Proof:

This transform pair follows directly from (8.63). The $f(t) \leftrightarrow F(\omega)$ correspondence is also shown in Figure 8.3.

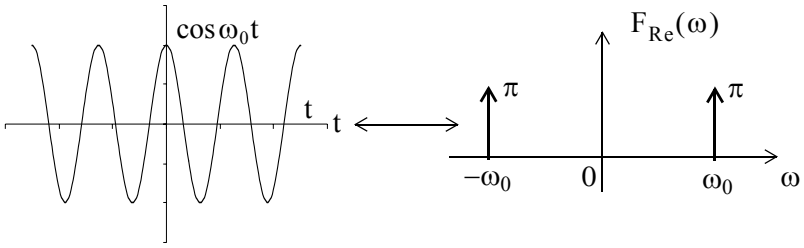


Figure 8.3. The Fourier transform of $f(t) = \cos\omega_0 t$

We know that $\cos\omega_0 t$ is real and even function of time, and we found out that its Fourier transform is a real and even function of frequency. This is consistent with the result in Table 8.7.

8.4.4 The Sine Function Pair

$$\sin\omega_0 t = \frac{1}{j2}(e^{j\omega_0 t} - e^{-j\omega_0 t}) \Leftrightarrow j\pi\delta(\omega-\omega_0) - j\pi\delta(\omega+\omega_0) \tag{8.65}$$

Proof:

This transform pair also follows directly from (8.63). The $f(t) \leftrightarrow F(\omega)$ correspondence is also shown in Figure 8.4.

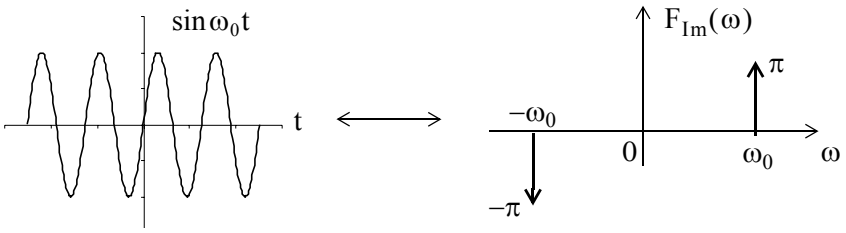


Figure 8.4. The Fourier transform of $f(t) = \sin\omega_0 t$

We know that $\sin\omega_0 t$ is real and odd function of time, and we found out that its Fourier transform is an imaginary and odd function of frequency. This is consistent with the result in Table 8.7.

8.4.5 The Signum Function Pair

$$\text{sgn}(t) = u_0(t) - u_0(-t) \Leftrightarrow \frac{2}{j\omega} \tag{8.66}$$

where $\text{sgn}(t)$ denotes the *signum function* shown in Figure 8.5.

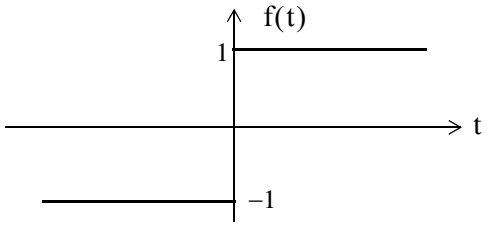


Figure 8.5. The signum function

Proof:

To derive the Fourier transform of the $\text{sgn}(t)$ function, it is convenient to express it as an exponential that approaches a limit as shown in Figure 8.6.

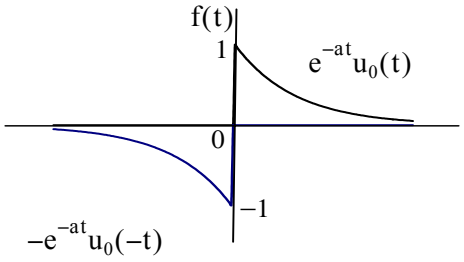


Figure 8.6. The signum function as an exponential approaching a limit

Then,

$$\text{sgn}(t) = \lim_{a \to 0} [e^{-at}u_0(t) - e^{at}u_0(-t)] \tag{8.67}$$

and

$$\begin{aligned}
\mathcal{F}\{\text{sgn}(t)\} &= \lim_{a \to 0} \left[\int_{-\infty}^{0} -e^{at}e^{-j\omega t}dt + \int_{0}^{\infty} e^{-at}e^{-j\omega t}dt \right] \\
&= \lim_{a \to 0} \left[\int_{-\infty}^{0} -e^{(a-j\omega)t}dt + \int_{0}^{\infty} e^{-(a+j\omega)t}dt \right] \\
&= \lim_{a \to 0} \left[\frac{-1}{a-j\omega} + \frac{1}{a+j\omega} \right] = \frac{-1}{-j\omega} + \frac{1}{j\omega} = \frac{2}{j\omega}
\end{aligned} \tag{8.68}$$

The $f(t) \leftrightarrow F(\omega)$ correspondence is also shown in Figure 8.7.

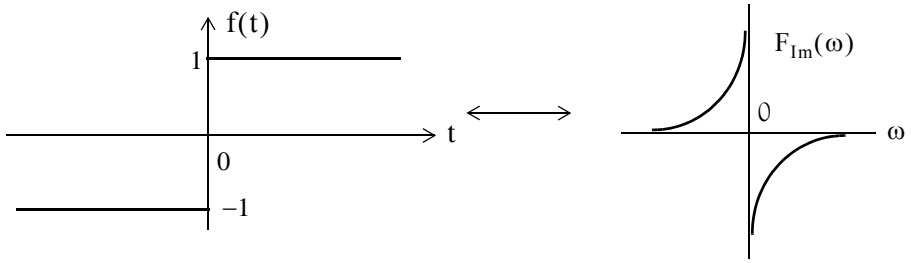


Figure 8.7. The Fourier transform of $\text{sgn}(t)$

We now know that $\text{sgn}(t)$ is real and odd function of time, and we found out that its Fourier transform is an imaginary and odd function of frequency. This is consistent with the result in Table 8.7.

8.4.6 The Unit Step Function Pair

$$\boxed{u_0(t) \Leftrightarrow \pi\delta(\omega) + \frac{1}{j\omega}} \tag{8.69}$$

Proof:

If we attempt to verify the transform pair of (8.69) by direct application of the Fourier transform definition, we will find that

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt = F(\omega) = \int_{0}^{\infty} e^{-j\omega t}dt = \frac{e^{-j\omega t}}{-j\omega}\bigg|_0^{\infty} \tag{8.70}$$

but we cannot say that $e^{-j\omega t}$ approaches 0 as $t \to \infty$, because $e^{-j\omega t} = 1\angle -\omega t$, that is, the magnitude of $e^{-j\omega t}$ is always unity, and its angle changes continuously as t assumes greater and greater values. Since the upper limit cannot be evaluated, the integral of (8.70) does not converge.

To work around this problem, we will make use of the $\text{sgn}(t)$ function which we express as

$$\boxed{\text{sgn}(t) = 2u_0(t) - 1} \tag{8.71}$$

This expression is derived from the waveform of Figure 8.8 below.

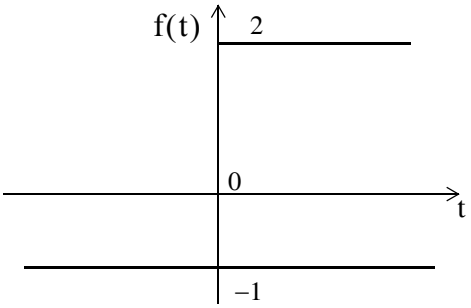


Figure 8.8. Alternate expression for the signum function

We rewrite (8.71) as

$$u_0(t) = \frac{1}{2}[1 + \text{sgn}(t)] = \frac{1}{2} + \frac{1}{2}\text{sgn}(t) \tag{8.72}$$

and since we know that $1 \Leftrightarrow 2\pi\delta(\omega)$ and $\text{sgn}(t) \Leftrightarrow 2/(j\omega)$, by substitution of these into (8.72) we obtain

$$u_0(t) \Leftrightarrow \pi\delta(\omega) + \frac{1}{j\omega}$$

and this is the same as(8.69). This is a complex function in the frequency domain whose real part is $\pi\delta(\omega)$ and imaginary part $-1/\omega$.

The $f(t) \leftrightarrow F(\omega)$ correspondence is also shown in Figure 8.9.

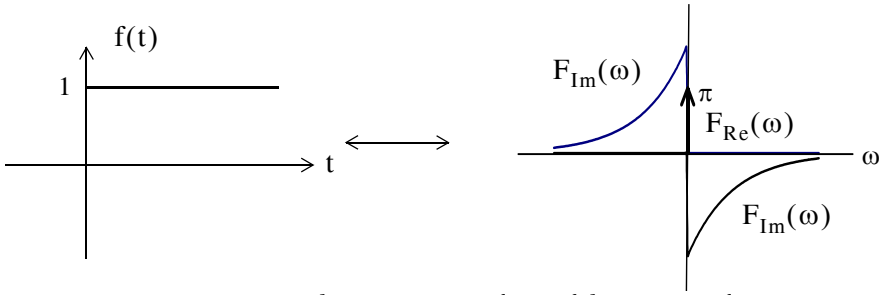


Figure 8.9. The Fourier transform of the unit step function

Since $u_0(t)$ is real but neither even nor odd function of time, its Fourier transform is a complex function of frequency as shown in (8.69). This is consistent with the result in Table 8.7.

Now, we will prove the time integration property of (8.49), Page 8–13, that is,

$$\int_{-\infty}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(\omega)}{j\omega} + \pi F(0)\delta(\omega)$$

as follows:

By the convolution integral,

$$u_0(t)*f(t) = \int_{-\infty}^{t} f(\tau)u_0(t-\tau)d\tau$$

and since $u_0(t-\tau) = 1$ for $t > \tau$, and it is zero otherwise, the above integral reduces to

$$u_0(t)*f(t) = \int_{-\infty}^{t} f(\tau)d\tau$$

Next, by the time convolution property,

$$u_0(t)*f(t) \Leftrightarrow U_0(\omega) \cdot F(\omega)$$

and since

$$U_0(\omega) = \pi\delta(\omega) + \frac{1}{j\omega}$$

using these results and the sampling property of the delta function, we obtain

$$U_0(\omega) \cdot F(\omega) = \left(\pi\delta(\omega) + \frac{1}{j\omega}\right)F(\omega) = = \pi\delta(\omega)F(\omega) + \frac{F(\omega)}{j\omega} = \pi F(0)\delta(\omega) + \frac{F(\omega)}{j\omega}$$

Thus, the time integration property is proved.

8.4.7 The $e^{-j\omega_0 t}u_0(t)$ Function Pair

$$e^{-j\omega_0 t}u_0(t) \Leftrightarrow \pi\delta(\omega-\omega_0)+\frac{1}{j(\omega-\omega_0)} \tag{8.73}$$

Proof:

From the Fourier transform of the unit step function,

$$u_0(t) \Leftrightarrow \pi\delta(\omega)+\frac{1}{j\omega}$$

and the frequency shifting property,

$$e^{j\omega_0 t}f(t) \Leftrightarrow F(\omega-\omega_0)$$

we obtain (8.73).

8.4.8 The $(\cos\omega_0 t)u_0(t)$ Function Pair

$$\begin{aligned}(\cos\omega_0 t)(u_0 t) &\Leftrightarrow \frac{\pi}{2}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]+\frac{1}{2j(\omega-\omega_0)}+\frac{1}{2j(\omega+\omega_0)}\\ &\Leftrightarrow \frac{\pi}{2}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]+\frac{j\omega}{\omega_0^2-\omega^2}\end{aligned} \tag{8.74}$$

Proof:

We express the cosine function as

$$\cos\omega_0 t = \frac{1}{2}(e^{j\omega_0 t}+e^{-j\omega_0 t})$$

From (8.73),

$$e^{-j\omega_0 t}u_0(t) \Leftrightarrow \pi\delta(\omega-\omega_0)+\frac{1}{j(\omega-\omega_0)}$$

and

$$e^{j\omega_0 t}u_0(t) \Leftrightarrow \pi\delta(\omega+\omega_0)+\frac{1}{j(\omega+\omega_0)}$$

Now, using

$$u_0(t) \Leftrightarrow \pi\delta(\omega)+\frac{1}{j\omega}$$

we obtain (8.74).

8.4.9 The $(\sin\omega_0 t)u_0(t)$ Function Pair

$$\boxed{(\sin\omega_0 t)(u_0 t) \Leftrightarrow \frac{\pi}{j2}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)] + \frac{\omega^2}{\omega_0^2-\omega^2}} \tag{8.75}$$

Proof:

We express the sine function as

$$\sin\omega_0 t = \frac{1}{j2}(e^{j\omega_0 t} - e^{-j\omega_0 t})$$

From (8.73),

$$e^{-j\omega_0 t}u_0(t) \Leftrightarrow \pi\delta(\omega-\omega_0) + \frac{1}{j(\omega-\omega_0)}$$

and

$$e^{j\omega_0 t}u_0(t) \Leftrightarrow \pi\delta(\omega+\omega_0) + \frac{1}{j(\omega+\omega_0)}$$

Using

$$u_0(t) \Leftrightarrow \pi\delta(\omega) + \frac{1}{j\omega}$$

we obtain (8.75).

8.5 Derivation of the Fourier Transform from the Laplace Transform

If a time function $f(t)$ is zero for $t \le 0$, we can obtain the Fourier transform of $f(t)$ from the one-sided Laplace transform of $f(t)$ by substitution of s with $j\omega$.

Example 8.1

It is known that $\mathcal{L}\,[e^{-\alpha t}u_0(t)] = \dfrac{1}{s+\alpha}$. Compute $\mathcal{F}\,\{e^{-\alpha t}u_0(t)\}$

Solution:

$$\mathcal{F}\,\{e^{-\alpha t}u_0(t)\} = \mathcal{L}\,[e^{-\alpha t}u_0(t)]\Big|_{s=j\omega} = \frac{1}{s+\alpha}\Big|_{s=j\omega} = \frac{1}{j\omega+\alpha}$$

Thus, we have obtained the following Fourier transform pair.

$$\boxed{e^{-\alpha t}u_0(t) \Leftrightarrow \frac{1}{j\omega+\alpha}} \tag{8.76}$$

Example 8.2

It is known that

$$\mathcal{L}\,[(e^{-\alpha t}\cos\omega_0 t)u_0(t)] = \frac{s+\alpha}{(s+\alpha)^2+\omega_0^2}$$

Compute $\mathcal{F}\,\{(e^{-\alpha t}\cos\omega_0 t)u_0(t)\}$

Solution:

$$\mathcal{F}\,\{(e^{-\alpha t}\cos\omega_0 t)\,u_0(t)\} = \mathcal{L}\,[(e^{-\alpha t}\cos\omega_0 t)u_0(t)]\Big|_{s=j\omega}$$

$$= \frac{s+\alpha}{(s+\alpha)^2+\omega_0^2}\Bigg|_{s=j\omega} = \frac{j\omega+\alpha}{(j\omega+\alpha)^2+\omega_0^2}$$

Thus, we have obtained the following Fourier transform pair.

$$\boxed{(e^{-\alpha t}\cos\omega_0 t)u_0(t) \Leftrightarrow \frac{j\omega+\alpha}{(j\omega+\alpha)^2+\omega_0^2}} \tag{8.77}$$

We can also find the Fourier transform of a time function $f(t)$ that has non–zero values for $t<0$, and it is zero for all $t>0$. But because the one–sided Laplace transform does not exist for $t<0$, we must first express the negative time function in the $t>0$ domain, and compute the one–sided Laplace transform. Then, the Fourier transform of $f(t)$ can be found by substituting s with $-j\omega$. In other words, when $f(t) = 0$ for $t \geq 0$, and $f(t) \neq 0$ for $t<0$, we use the substitution

$$\boxed{\mathcal{F}\,\{f(t)\} = \mathcal{L}\,[f(-t)]\Big|_{s=-j\omega}} \tag{8.78}$$

Example 8.3

Compute the Fourier transform of $f(t) = e^{-a|t|}$

a. using the Fourier transform definition

b. by substitution into the Laplace transform equivalent

Solution:

a. Using the Fourier transform definition, we obtain

$$\mathcal{F}\{e^{-a|t|}\} = \int_{-\infty}^{0} e^{at}e^{-j\omega t}dt + \int_{0}^{\infty} e^{-at}e^{-j\omega t}dt = \int_{-\infty}^{0} e^{(a-j\omega)t}dt + \int_{0}^{\infty} e^{-(a+j\omega)t}dt$$

$$= \frac{1}{a-j\omega} + \frac{1}{a+j\omega} = \frac{2}{\omega^2 + a^2}$$

and thus we have the transform pair

$$\boxed{e^{-a|t|} \Leftrightarrow \frac{2}{\omega^2 + a^2}} \tag{8.79}$$

b.By substitution into the Laplace transform equivalent, we obtain

$$\mathcal{F}\{e^{-a|t|}\} = \mathcal{L}[e^{-at}]\Big|_{s=j\omega} + \mathcal{L}[e^{at}]\Big|_{s=-j\omega} = \frac{1}{s+a}\Big|_{s=j\omega} + \frac{1}{s+a}\Big|_{s=-j\omega}$$

$$= \frac{1}{j\omega + a} + \frac{1}{-j\omega + a} = \frac{2}{\omega^2 + a^2}$$

and this result is the same as (8.79). We observe that since $f(t)$ is real and even, $F(\omega)$ is also real and even.

8.6 Fourier Transforms of Common Waveforms

In this section, we will derive the Fourier transform of some common time domain waveforms. These are described in Subsections 8.6.1 through 8.6.6 below.

8.6.1 The Transform of $f(t) = A[u_0(t+T) - u_0(t-T)]$

The symmetric rectangular pulse waveform

$$f(t) = A[u_0(t+T) - u_0(t-T)] \tag{8.80}$$

is shown in Figure 8.10.

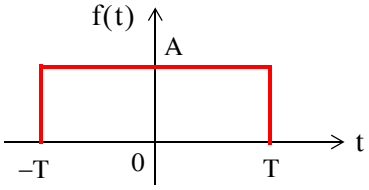


Figure 8.10. Rectangular pulse waveform $f(t) = A[u_0(t+T) - u_0(t-T)]$

Using the definition of the Fourier transform, we obtain

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt = \int_{-T}^{T} Ae^{-j\omega t}dt = \frac{Ae^{-j\omega t}}{-j\omega}\Bigg|_{-T}^{T}$$

$$= \frac{Ae^{-j\omega t}}{j\omega}\Bigg|_{T}^{-T} = \frac{A(e^{j\omega t} - e^{-j\omega t})}{j\omega} = 2A\frac{\sin\omega T}{\omega} = 2AT\frac{\sin\omega T}{\omega T}$$

We observe that the transform of this pulse has the $\sin x/x$ form, and has its maximum value $2AT$ at $\omega T = 0$.* Thus, we have the waveform pair

$$\boxed{A[u_0(t+T) - u_0(t-T)] \Leftrightarrow 2AT\frac{\sin\omega T}{\omega T}} \tag{8.81}$$

The $f(t) \leftrightarrow F(\omega)$ correspondence is also shown in Figure 8.11, where we observe that the ω axis crossings occur at values of $\omega T = \pm n\pi$ where n is an integer.

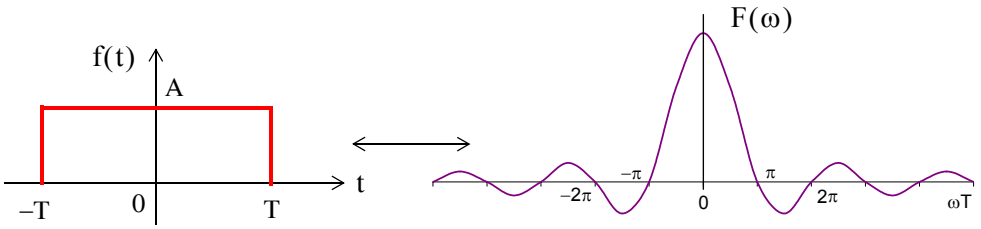


Figure 8.11. The waveform $f(t) = A[u_0(t+T) - u_0(t-T)]$ *and its Fourier transform*

We also observe that since $f(t)$ is real and even, $F(\omega)$ is also real and even.

8.6.2 The Transform of $f(t) = A[u_0(t) - u_0(t-2T)]$

The shifted–to–the–right rectangular waveform

$$f(t) = A[u_0(t) - u_0(t-2T)] \tag{8.82}$$

is shown in Figure 8.12.

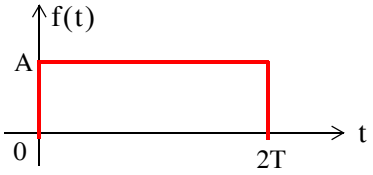


Figure 8.12. Pulse for $f(t) = A[u_0(t) - u_0(t-2T)]$

* *We recall that* $\lim_{x\to 0}\frac{\sin x}{x} = 1$

Using the definition of the Fourier transform, we obtain

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt = \int_{0}^{2T} Ae^{-j\omega t}dt = \frac{Ae^{-j\omega t}}{-j\omega}\bigg|_{0}^{2T} = \frac{Ae^{-j\omega t}}{j\omega}\bigg|_{2T}^{0} = \frac{A(1-e^{-j\omega 2T})}{j\omega}$$

and making the substitutions

$$\begin{aligned} 1 &= e^{j\omega T}\cdot e^{-j\omega T} \\ e^{-j\omega 2T} &= e^{-j\omega T}\cdot e^{-j\omega T} \end{aligned} \tag{8.83}$$

we obtain

$$F(\omega) = \frac{Ae^{-j\omega T}}{\omega}\left(\frac{e^{j\omega T}-e^{-j\omega T}}{j}\right) = 2Ae^{-j\omega T}\left(\frac{\sin\omega T}{\omega}\right) = 2ATe^{-j\omega T}\left(\frac{\sin\omega T}{\omega T}\right) \tag{8.84}$$

Alternate Derivation:

We can obtain the Fourier transform of (8.82) using the time shifting property, i.e,

$$f(t-t_0) \Leftrightarrow F(\omega)e^{-j\omega t_0}$$

and the result of Subsection 8.6.1. Thus, multiplying $2AT\,\frac{\sin\omega T}{\omega T}$ by $e^{-j\omega T}$, we obtain (8.84).

We observe that $F(\omega)$ is complex* since $f(t)$ is neither an even nor an odd function.

8.6.3 The Transform of $f(t) = A[u_0(t+T) + u_0(t) - u_0(t-T) - u_0(t-2T)]$

The waveform

$$f(t) = A[u_0(t+T) + u_0(t) - u_0(t-T) - u_0(t-2T)] \tag{8.85}$$

is shown in Figure 8.13.

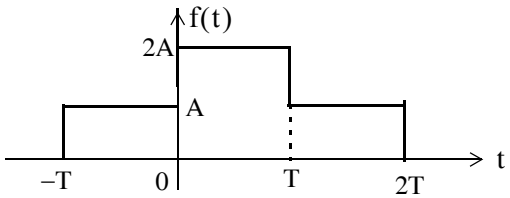


Figure 8.13. Waveform for $f(t) = A[u_0(t+T) + u_0(t) - u_0(t-T) - u_0(t-2T)]$

We observe that this waveform is the sum of the waveforms of Subsections 8.6.1 and 8.6.2. We also observe that this waveform is obtained by the graphical addition of the waveforms of Figures

* *We recall that* $e^{-j\omega T}$ *consists of a real and an imaginary part.*

8.10 and 8.12. Therefore, we will apply the linearity property to obtain the Fourier transform of this waveform.

We denote the transforms of Subsections 8.6.1 and 8.6.2 as $F_1(\omega)$ and $F_2(\omega)$ respectively, and we obtain

$$\begin{aligned} F(\omega) &= F_1(\omega) + F_2(\omega) = 2AT\,\frac{\sin\omega T}{\omega T} + 2ATe^{-j\omega T}\left(\frac{\sin\omega T}{\omega T}\right) \\ &= 2AT(1+e^{-j\omega T})\frac{\sin\omega T}{\omega T} = 2ATe^{-j\frac{\omega T}{2}}\left(e^{j\frac{\omega T}{2}} + e^{-j\frac{\omega T}{2}}\right)\frac{\sin\omega T}{\omega T} \\ &= 4ATe^{-j\frac{\omega T}{2}}\cos\left(\frac{\omega T}{2}\right)\frac{\sin\omega T}{\omega T} \end{aligned} \tag{8.86}$$

We observe that $F(\omega)$ is complex since $f(t)$ of (8.85) is neither an even nor an odd function.

8.6.4 The Transform of $f(t) = A\cos\omega_0 t[u_0(t+T) - u_0(t-T)]$

The transform of the waveform

$$f(t) = A\cos\omega_0 t[u_0(t+T) - u_0(t-T)] \tag{8.87}$$

can be obtained from (8.45), Page 8–12, that is,

$$f(t)\cos\omega_0 t \Leftrightarrow \frac{F(\omega-\omega_0) + F(\omega+\omega_0)}{2}$$

and from (8.81), Page 8–28, that is,

$$A[u_0(t+T) - u_0(t-T)] \Leftrightarrow 2AT\,\frac{\sin\omega T}{\omega T}$$

Then,

$$\boxed{A\cos\omega_0 t[u_0(t+T) - u_0(t-T)] \Leftrightarrow AT\left[\frac{\sin[(\omega-\omega_0)T]}{(\omega-\omega_0)T} + \frac{\sin[(\omega+\omega_0)T]}{(\omega+\omega_0)T}\right]} \tag{8.88}$$

We also observe that since $f(t)$ is real and even, $F(\omega)$ is also real and even.*

* *The* $\sin x/x$ *is an even function.*

8.6.5 The Transform of a Periodic Time Function with Period T

The Fourier transform of a periodic time function with period T can be derived from the definition of the exponential Fourier series, that is,

$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t} \tag{8.89}$$

where $\omega_0 = 2\pi/T$, and from (8.63), Page 8-19, that is,

$$e^{j\omega_0 t} \Leftrightarrow 2\pi\delta(\omega - \omega_0)$$

Then,

$$\begin{aligned} C_n e^{j\omega_0 t} &\Leftrightarrow 2\pi C_1 \delta(\omega - \omega_0) \\ C_n e^{j2\omega_0 t} &\Leftrightarrow 2\pi C_2 \delta(\omega - 2\omega_0) \\ &\cdots \\ C_n e^{jn\omega_0 t} &\Leftrightarrow 2\pi C_n \delta(\omega - n\omega_0) \end{aligned} \tag{8.90}$$

Taking the Fourier transform of (8.89), and applying the linearity property for the transforms of (8.90), we obtain

$$\mathcal{F}\{f(t)\} = \mathcal{F}\left\{\sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t}\right\} = \sum_{n=-\infty}^{\infty} C_n \mathcal{F}\{e^{jn\omega_0 t}\} = 2\pi \sum_{n=-\infty}^{\infty} C_n \delta(\omega - n\omega_0) \tag{8.91}$$

The line spectrum of the Fourier transform of (8.91) is shown in Figure 8.14.

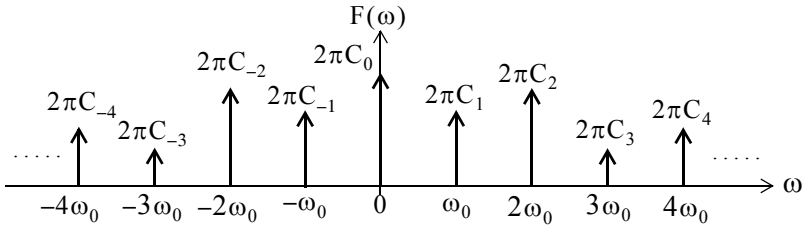


Figure 8.14. Line spectrum for relation (8.91)

The line spectrum of Figure 8.14 reveals that the Fourier transform of a periodic time function with period T, consists of a train of equally spaced delta functions. The strength of each $\delta(\omega - n\omega_0)$ is equal to 2π times the coefficient C_n.

8.6.6 The Transform of the Periodic Time Function $f(t) = A\sum_{n=-\infty}^{\infty}\delta(t-nT)$

The periodic time function

$$f(t) = A\sum_{n=-\infty}^{\infty}\delta(t-nT) \tag{8.92}$$

consists of a train of equally spaced delta functions in the time domain, and each has the same strength A, as shown in Figure 8.15.

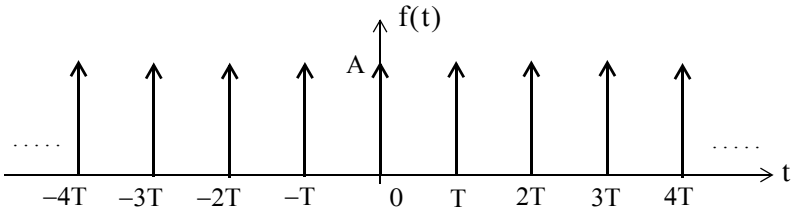


Figure 8.15. Waveform for $f(t) = A\sum_{n=-\infty}^{\infty}\delta(t-nT)$

Since this is a periodic function of time, its Fourier transform is as derived in Subsection 8.6.5, that is, relation (8.91). Then,

$$F(\omega) = 2\pi\sum_{n=-\infty}^{\infty}C_n\delta(\omega-n\omega_0) \tag{8.93}$$

where $\omega_0 = 2\pi/T$, and C_n is found from the exponential Fourier series

$$C_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)e^{-jn\omega_0 t}dt \tag{8.94}$$

From the waveform of Figure 8.15, we observe that, within the limits of integration from $-T/2$ to $+T/2$, there is only the impulse $\delta(t)$ at the origin. Therefore, replacing $f(t)$ with $\delta(t)$ and using the sifting property of the delta function, we obtain

$$C_n = \frac{1}{T}\int_{-T/2}^{T/2}\delta(t)e^{-jn\omega_0 t}dt = \frac{1}{T} \tag{8.95}$$

Thus, we see that all C_n coefficients are equal to $1/T$, and (8.93) is expressed as

$$F(\omega) = \frac{2\pi}{T}\sum_{n=-\infty}^{\infty}\delta(\omega-n\omega_0) \tag{8.96}$$

The Fourier transform of the waveform of Figure 8.15 is shown in Figure 8.16.

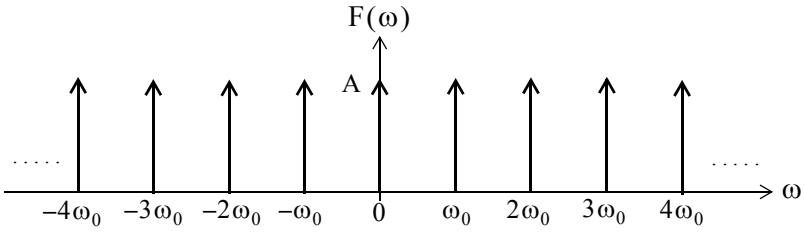


Figure 8.16. The Fourier transform of a train of equally spaced delta functions

Figure 8.16 shows that the Fourier transform of a periodic train of equidistant delta functions in the time domain, is a periodic train of equally spaced delta functions in the frequency domain. This result is the basis for the proof of the *sampling theorem* which states that a time function f(t) can be uniquely determined from its values at a sequence of equidistant points in time.

8.7 Using MATLAB for Finding the Fourier Transform of Time Functions

MATLAB has the built–in **fourier** and **ifourier** functions to compute the Fourier transform and its inverse. Their descriptions and examples, can be displayed with the **help fourier** and **help ifourier** commands. In examples 8.4 through 8.7 we present some Fourier transform pairs, and how they are verified with MATLAB.

Example 8.4

$$e^{-\frac{1}{2}t^2} \Leftrightarrow \sqrt{2\pi}e^{-\frac{1}{2}\omega^2} \tag{8.97}$$

This time function, like the time function of Subsection 8.6.6, is its own Fourier transform multiplied by the constant $\sqrt{2\pi}$.

```
syms t v w x; ft=exp(–t^2/2); Fw=fourier(ft)

Fw =
  2^(1/2)*pi^(1/2)*exp(-1/2*w^2)

pretty(Fw)

                 1/2   1/2             2
                2     pi     exp(- 1/2 w )

% Check answer by computing the Inverse using "ifourier"
ft=ifourier(Fw)

ft =
  exp(-1/2*x^2)
```

Example 8.5

$$te^{-t^2} \Leftrightarrow \frac{1}{2}j\sqrt{\pi}\omega e^{-\frac{1}{4}\omega^2} \tag{8.98}$$

```
syms t v w x; ft=t*exp(–t^2); Fw=fourier (ft)

   Fw =
     -1/2*i*pi^(1/2)*w*exp(-1/4*w^2)

pretty(Fw)

                      1/2                 2
          - 1/2 i pi     w exp(- 1/4 w )
```

Example 8.6

$$-e^{-t}u_0(t) + 3\delta(t) \Leftrightarrow -\frac{1}{j\omega + 1} + 3 \tag{8.99}$$

```
syms t v w x; fourier(sym('–exp(–t)*Heaviside(t)+3*Dirac(t)'))

   ans =
     -1/(1+i*w)+3
```

Example 8.7

$$u_0(t) \Leftrightarrow \pi\delta(\omega) + \frac{1}{j\omega} \tag{8.100}$$

```
syms t v w x; u0=sym('Heaviside(t)'); Fw=fourier(u0)

   Fw =
   pi*Dirac(w)-i/w
```

We have summarized the most common Fourier transform pairs in Table 8.9.

8.8 The System Function and Applications to Circuit Analysis

We recall from Chapter 6 that, by definition, the convolution integral is

$$h(t)*u(t) = \int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau \tag{8.101}$$

We let

$$g(t) = f(t)*h(t) \tag{8.102}$$

TABLE 8.9 Common Fourier transform pairs

f(t)	**F(ω)**
$\delta(t)$	1
$\delta(t-t_0)$	$e^{-j\omega t_0}$
1	$2\pi\delta(\omega)$
$e^{-j\omega t_0}$	$2\pi\delta(\omega-\omega_0)$
$\text{sgn}(t)$	$2/(j\omega)$
$u_0(t)$	$\frac{1}{j\omega}+\pi\delta(\omega)$
$\cos\omega_0 t$	$\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$
$\sin\omega_0 t$	$j\pi[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)]$
$e^{-at}u_0(t)$ $a>0$	$\frac{1}{j\omega+a}$ $a>0$
$te^{-at}u_0(t)$ $a>0$	$\frac{1}{(j\omega+a)^2}$ $a>0$
$e^{-at}\cos\omega_0 t u_0(t)$ $a>0$	$\frac{j\omega+a}{(j\omega+a)^2+\omega^2}$ $a>0$
$e^{-at}\sin\omega_0 t u_0(t)$ $a>0$	$\frac{\omega}{(j\omega+a)^2+\omega^2}$ $a>0$
$A[u_0(t+T)-u_0(t-T)]$	$2AT\frac{\sin\omega T}{\omega T}$

and recalling that convolution in the time domain corresponds to multiplication in the frequency domain, we obtain

$$f(t)*h(t) = g(t) \Leftrightarrow G(\omega) = F(\omega)\cdot H(\omega) \tag{8.103}$$

We call $H(\omega)$ the *system function*. From (8.103), we see that the system function $H(\omega)$ and the impulse response $h(t)$ form the Fourier transform pair

$$h(t) \Leftrightarrow H(\omega) \tag{8.104}$$

Therefore, if we know the impulse response $h(t)$, we can compute the response $g(t)$ of any input $f(t)$, by multiplication of the Fourier transforms $H(\omega)$ and $F(\omega)$ to obtain $G(\omega)$. Then, we take the Inverse Fourier transform of $G(\omega)$ to obtain the response $g(t)$.

Example 8.8

For the linear network of Figure 8.17 (a) below, it is known that the impulse response is as shown in Figure 8.17 (b). Use the Fourier transform method to compute the response $g(t)$ when the input $f(t)$ is as shown in Figure 8.17 (c).

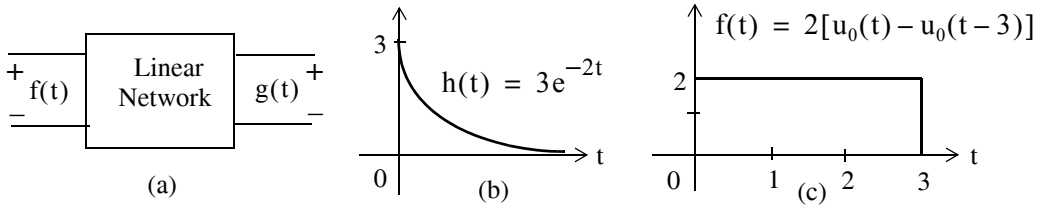


Figure 8.17. Figure for Example 8.8.

Solution:

To facilitate the computations, we denote the input as $f(t) = f_1(t) + f_2(t)$ where

$$f_1(t) = 2u_0(t)$$

and

$$f_2(t) = -2u_0(t-3)$$

The system function $H(\omega)$ is the Fourier transform of the impulse response $h(t)$. Thus,

$$\mathcal{F}\{h(t)\} = H(\omega) = \frac{3}{j\omega + 2}$$

Let $F_1(\omega)$ be the Fourier transform of $f_1(t)$, that is,

$$\mathcal{F}\{f_1(t)\} = F_1(\omega) = 2\left(\pi\delta(\omega) + \frac{1}{j\omega}\right)$$

Then,

$$G_1(\omega) = H(\omega) \cdot F_1(\omega) = \frac{3}{j\omega + 2} \cdot 2\left(\pi\delta(\omega) + \frac{1}{j\omega}\right)$$

or

$$G_1(\omega) = \frac{6\pi}{j\omega + 2}\delta(\omega) + \frac{3}{j\omega(j\omega + 2)} \tag{8.105}$$

To evaluate the first term of (8.105), we apply the sampling property of the delta function, i.e.,

$$X(\omega) \cdot \delta(\omega) = X(0) \cdot \delta(\omega)$$

and this term reduces to $3\pi\delta(\omega)$ or $1.5[2\pi\delta(\omega)]$. Since $1 \Leftrightarrow 2\pi\delta(\omega)$, the Inverse Fourier transform of this term is

$$\mathcal{F}^{-1}\{1.5[2\pi\delta(\omega)]\} = 1.5 \tag{8.106}$$

To find the Inverse Fourier transform of the second term in (8.105), we use partial fraction expansion. Thus,

$$\frac{3}{j\omega(j\omega + 2)} = \frac{1.5}{j\omega} - \frac{1.5}{(j\omega + 2)}$$

and therefore,

$$g_1(t) = 1.5 + \mathcal{F}^{-1}\left\{\frac{1.5}{j\omega} - \frac{1.5}{(j\omega + 2)}\right\} = 1.5 + \mathcal{F}^{-1}\left\{0.75\frac{2}{j\omega} - \frac{1.5}{(j\omega + 2)}\right\}$$

or

$$\begin{aligned} g_1(t) &= 1.5 + 0.75\,\mathrm{sgn}(t) - 1.5e^{-2t}u_0(t) \\ &= 1.5 + 0.75[2u_0(t) - 1] - 1.5e^{-2t}u_0(t) = 0.75 + 1.5(1 - e^{-2t})u_0(t) \end{aligned} \tag{8.107}$$

Next, we denote the response due to the second term of the input as $g_2(t)$, and replacing $u_0(t)$ in (8.107) with $u_0(t-3)$, we obtain

$$g_2(t) = 0.75 + 1.5(1 - e^{-2(t-3)})u_0(t-3) \tag{8.108}$$

Now, we combine (8.107) with (8.108), and we obtain

$$g(t) = g_1(t) - g_2(t)$$

or

$$g(t) = 1.5\{(1 - e^{-2t})u_0(t) - (1 - e^{-2(t-3)})u_0(t-3)\}$$

Example 8.9

For the circuit of Figure 8.18, use the Fourier transform method, and the system function $H(\omega)$ to compute $v_L(t)$. Assume $i_L(0^-)$.

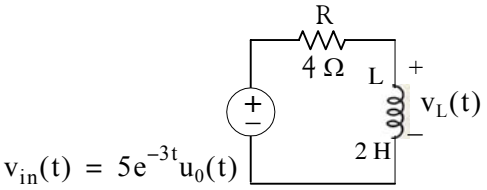


Figure 8.18. Circuit for Example 8.9

Solution:

We will find the system function $H(\omega)$ from the phasor equivalent circuit shown in Figure 8.19.

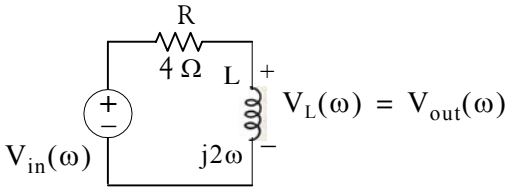


Figure 8.19. Phasor circuit for Example 8.9

From the phasor circuit of Figure 8.19,

$$V_{out}(\omega) = \frac{j2\omega}{4+j2\omega}V_{in}(\omega) = \frac{j\omega}{j\omega+2}V_{in}(\omega)$$

and the system function is

$$H(\omega) = \frac{V_{out}(\omega)}{V_{in}(\omega)} = \frac{j\omega}{j\omega+2}$$

Also,

$$v_{in}(t) = 5e^{-3t}u_0(t) \Leftrightarrow V_{in}(\omega) = \frac{5}{j\omega+3}$$

Then,

$$V_{out}(\omega) = H(\omega)V_{in}(\omega) = \frac{j\omega}{j\omega+2}\cdot\frac{5}{j\omega+3} = \frac{r_1}{j\omega+2}+\frac{r_2}{j\omega+3}$$

and by partial fraction expansion, we find that $r_1 = -10$ and $r_2 = 15$. Thus,

$$V_{out}(\omega) = \frac{15}{j\omega+3}-\frac{10}{j\omega+2}$$

and

$$v_L(t) = v_{out}(t) = \mathcal{F}^{-1}\left\{\frac{15}{j\omega+3}-\frac{-10}{j\omega+2}\right\} = 15e^{-3t}-10e^{-2t}$$

or

$$v_L(t) = 5(3e^{-3t}-2e^{-2t})u_0(t) \tag{8.109}$$

Example 8.10

For the linear network of Figure 8.20, the input–output relationship is

$$\frac{d}{dt}v_{out}(t) + 4v_{out}(t) = 10v_{in}(t) \tag{8.110}$$

where $v_{in}(t)$ is as shown in Figure 8.20. Use the Fourier transform method, and the system function $H(\omega)$ to compute the output $v_{out}(t)$.

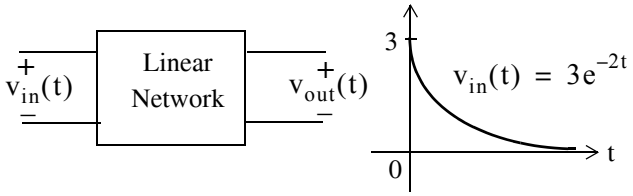


Figure 8.20. Network for Example 8.10

Solution:

Taking the Fourier transform of both sides of (8.110), and recalling that

$$\frac{d^n}{dt^n} f(t) \Leftrightarrow (j\omega)^n F(\omega)$$

we obtain,

$$j\omega V_{out}(\omega) + 4V_{out}(\omega) = 10V_{in}(\omega)$$

or

$$(j\omega + 4)V_{out}(\omega) = 10V_{in}(\omega)$$

and thus,

$$H(\omega) = \frac{V_{out}(\omega)}{V_{in}(\omega)} = \frac{10}{j\omega + 4} \tag{8.111}$$

Also,

$$V_{in}(\omega) = \mathcal{F}\{v_{in}(t)\} = \mathcal{F}(3e^{-2t}u_0(t)) = \frac{3}{j\omega + 2} \tag{8.112}$$

and

$$V_{out}(\omega) = H(\omega) \cdot V_{in}(\omega) = \frac{10}{j\omega + 4} \cdot \frac{3}{j\omega + 2} = \frac{r_1}{j\omega + 4} + \frac{r_2}{j\omega + 2}$$

By partial fraction expansion, we find that $r_1 = -15$ and $r_2 = 15$. Then,

$$V_{out}(\omega) = \frac{15}{j\omega + 2} + \frac{-15}{j\omega + 4} \tag{8.113}$$

Therefore,

$$v_{out}(t) = \mathcal{F}^{-1}\left\{\frac{15}{j\omega+2} + \frac{-15}{j\omega+4}\right\} = 15(e^{-2t} - e^{-4t})u_0(t) \tag{8.114}$$

Example 8.11

The voltage across an $1\ \Omega$ resistor is known to be $v_R(t) = 3e^{-2t}u_0(t)$. Compute the energy dissipated in this resistor for $0 < t < \infty$, and verify the result by application of Parseval's theorem.

Solution:

The instantaneous power absorbed by the resistor is

$$p_R = v_R^2/1 = v_R^2 = 9e^{-4t}u_0(t) \tag{8.115}$$

and thus, the energy is

$$W_R = \int_0^\infty v_R^2 dt = \int_0^\infty 9e^{-4t}dt = 9\frac{e^{-4t}}{-4}\Big|_0^\infty = \frac{9}{4}e^{-4t}\Big|_\infty^0 = 2.25 \text{ joules} \tag{8.116}$$

Parseval's theorem states that

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |F(\omega)|^2 d\omega \tag{8.117}$$

Since

$$F(\omega) = \mathcal{F}\{3e^{-2t}u_0(t)\} = \frac{3}{j\omega+2} \tag{8.118}$$

and

$$|F(\omega)|^2 = F(\omega)\cdot F^*(\omega) = \frac{9}{\omega^2+2^2} \tag{8.119}$$

by substitution into the right integral of (8.117) we obtain

$$W_R = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{9}{\omega^2+2^2}d\omega \tag{8.120}$$

We observe that the integrand of (8.120) is an even function of ω; therefore, we can multiply the integral by 2, and integrate from 0 to ∞. Then,

$$W_R = \frac{2}{2\pi}\int_0^\infty \frac{9}{\omega^2+2^2}d\omega = \frac{9}{\pi}\int_0^\infty \frac{1}{\omega^2+2^2}d\omega \tag{8.121}$$

From tables of integrals,

$$\int \frac{1}{a^2 + x^2} dx = \frac{1}{a} \operatorname{atan} \frac{x}{a} + C$$

Thus,

$$W_R = \frac{9}{\pi} \left(\frac{1}{2} \operatorname{atan} \frac{\omega}{2} \right) \Big|_0^{\infty} = \frac{9}{2\pi} \cdot \frac{\pi}{2} = 2.25 \text{ joules} \tag{8.122}$$

We observe that (8.122) is in agreement with (8.116).

8.9 Summary

- The Fourier transform is defined as

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$$

- The Inverse Fourier transform is defined as

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega$$

- The Fourier transform is, in general, a complex function. We can express it as the sum of its real and imaginary components, or in exponential form as

$$F(\omega) = \mathrm{Re}\{F(\omega)\} + j\mathrm{Im}\{F(\omega)\} = |F(\omega)|e^{j\varphi(\omega)}$$

- We often use the following notations to express the Fourier transform and its inverse.

$$\mathcal{F}\{f(t)\} = F(\omega)$$

$$\mathcal{F}^{-1}\{F(\omega)\} = f(t)$$

- If $f(t)$ is real, $F(\omega)$ is, in general, complex.
- If $f(t)$ is real and even, $F(\omega)$ is also real and even.
- If $f(t)$ is real and odd, $F(\omega)$ is imaginary and odd.
- If $f(t)$ is imaginary, $F(\omega)$ is, in general, complex.
- If $f(t)$ is imaginary and even, $F(\omega)$ is also imaginary and even.
- If $f(t)$ is imaginary and odd, $F(\omega)$ is real and odd.
- If $F(-\omega) = F^*(\omega)$, $f(t)$ is real.
- The linearity property states that

$$a_1 f_1(t) + a_2 f_2(t) + \ldots + a_n f_n(t) \Leftrightarrow a_1F_1(\omega) + a_2F_2(\omega) + \ldots + a_nF_n(\omega)$$

- The symmetry property states that

$$F(t) \Leftrightarrow 2\pi f(-\omega)$$

- The scaling property states that

$$f(at) \Leftrightarrow \frac{1}{|a|}F\left(\frac{\omega}{a}\right)$$

- The time shifting property states that

$$f(t-t_0) \Leftrightarrow F(\omega)e^{-j\omega t_0}$$

- The frequency shifting property states that

$$e^{j\omega_0 t}f(t) \Leftrightarrow F(\omega-\omega_0)$$

- The Fourier transforms of the modulated signals $f(t)\cos\omega t$ and $f(t)\sin\omega t$ are

$$f(t)\cos\omega_0 t \Leftrightarrow \frac{F(\omega-\omega_0)+F(\omega+\omega_0)}{2}$$

$$f(t)\sin\omega_0 t \Leftrightarrow \frac{F(\omega-\omega_0)-F(\omega+\omega_0)}{j2}$$

- The time differentiation property states that

$$\frac{d^n}{dt^n}f(t) \Leftrightarrow (j\omega)^n F(\omega)$$

- The frequency differentiation property states that

$$(-jt)^n f(t) \Leftrightarrow \frac{d^n}{d\omega^n}F(\omega)$$

- The time integration property states that

$$\int_{-\infty}^{t} f(\tau)d\tau \Leftrightarrow \frac{F(\omega)}{j\omega}+\pi F(0)\delta(\omega)$$

- If $F(\omega)$ is the Fourier transform of the complex function $f(t)$, then,

$$f^*(t) \Leftrightarrow F^*(-\omega)$$

- The time convolution property states that

$$f_1(t)*f_2(t) \Leftrightarrow F_1(\omega)F_2(\omega)$$

- The frequency convolution property states that

$$f_1(t)f_2(t) \Leftrightarrow \frac{1}{2\pi}F_1(\omega)*F_2(\omega)$$

- The area under a time function $f(t)$ is equal to the value of its Fourier transform evaluated at $\omega = 0$. In other words,

$$F(0) = \int_{-\infty}^{\infty} f(t)dt$$

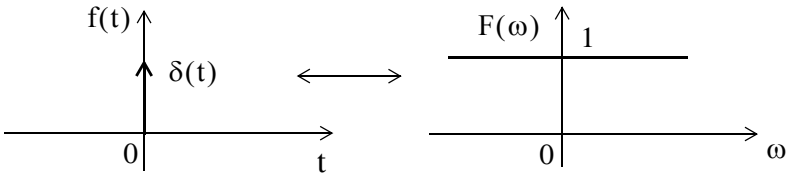
- The value of a time function $f(t)$, evaluated at $t = 0$, is equal to the area under its Fourier transform $F(\omega)$ times $1/2\pi$. In other words,

$$f(0) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)d\omega$$

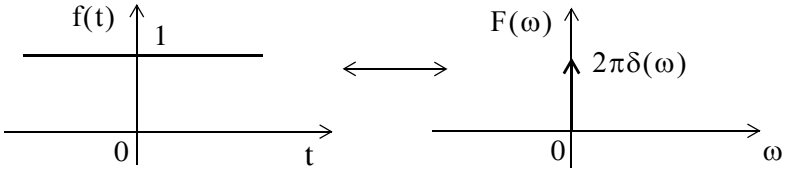
- Parseval's theorem states that

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |F(\omega)|^2 d\omega$$

- The delta function and its Fourier transform are as shown below.



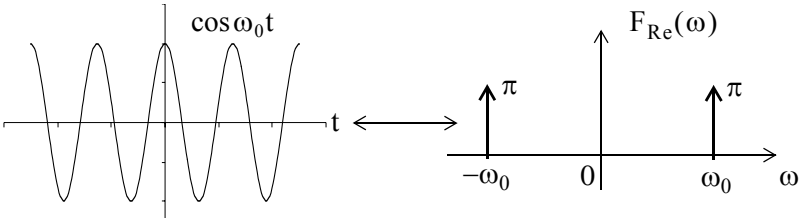
- The unity time function and its Fourier transform are as shown below.

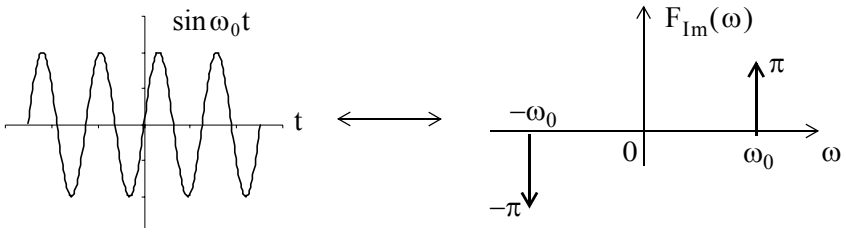


- The Fourier transform of the complex time function $e^{j\omega_0 t}$ is as indicated below.

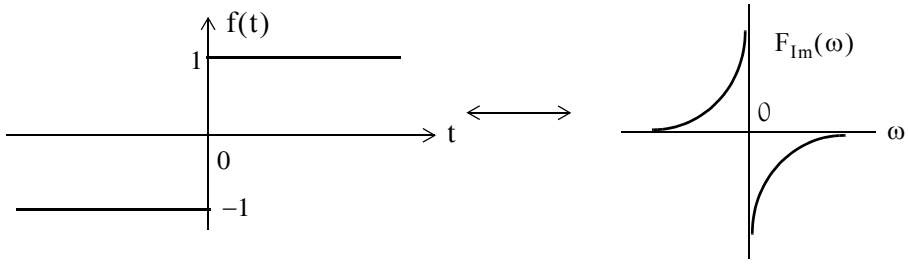
$$e^{j\omega_0 t} \Leftrightarrow 2\pi\delta(\omega - \omega_0)$$

- The Fourier transforms of the time functions $\cos\omega_0 t$, and $\sin\omega_0 t$ are as shown below.



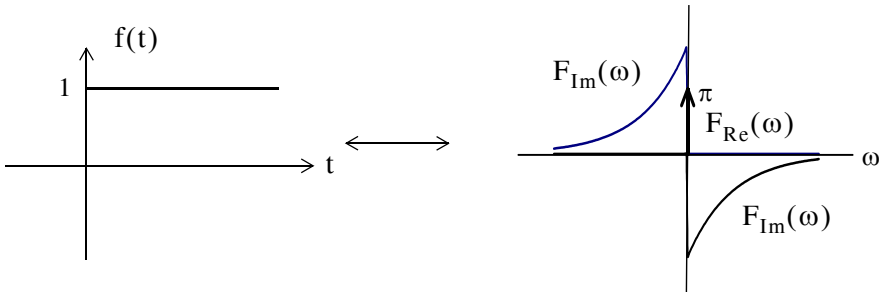


- The signum function and its Fourier transform are as shown below.



- The unit step function and its Fourier transform are as shown below.

.



- The Fourier transform pairs of $e^{-j\omega_0 t}u_0(t)$, $u_0(t)\cos\omega_0 t$, and $u_0(t)\sin\omega_0 t$ are as follows:

$$e^{-j\omega_0 t}u_0(t) \Leftrightarrow \pi\delta(\omega-\omega_0)+\frac{1}{j(\omega-\omega_0)}$$

$$u_0(t)\cos\omega_0 t \Leftrightarrow \frac{\pi}{2}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]+\frac{j\omega}{\omega_0^2-\omega^2}$$

$$u_0(t)\sin\omega_0 t \Leftrightarrow \frac{\pi}{j2}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]+\frac{\omega^2}{\omega_0^2-\omega^2}$$

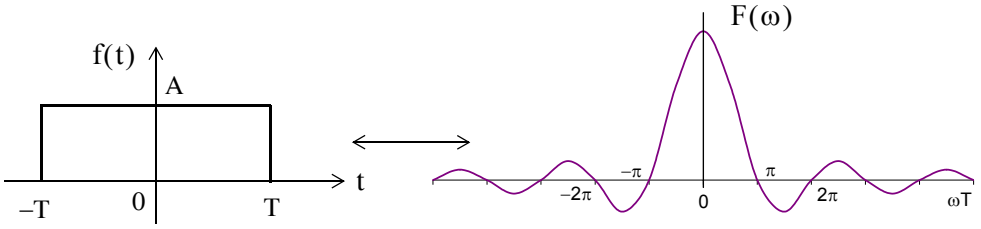
- If a time function $f(t)$ is zero for $t \le 0$, we can obtain the Fourier transform of $f(t)$ from the one-sided Laplace transform of $f(t)$ by substitution of s with $j\omega$.

- If a time function $f(t) = 0$ for $t \geq 0$, and $f(t) \neq 0$ for $t < 0$, we use the substitution

$$\mathcal{F}\{f(t)\} = \mathcal{L}[f(-t)]\big|_{s=-j\omega}$$

to obtain the Fourier transform of $f(t)$ from the one-sided Laplace transform of $f(t)$.

- The pulse function $f(t) = A[u_0(t+T) - u_0(t-T)]$ and its Fourier transform are as shown below.



- The Fourier transform of a periodic time function with period T is

$$\mathcal{F}\{f(t)\} = \mathcal{F}\left\{\sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t}\right\} = \sum_{n=-\infty}^{\infty} C_n \mathcal{F}\{e^{jn\omega_0 t}\} = 2\pi \sum_{n=-\infty}^{\infty} C_n \delta(\omega - n\omega_0)$$

- The Fourier transform of a periodic train of equidistant delta functions in the time domain, is a periodic train of equally spaced delta functions in the frequency domain.
- The system function $H(\omega)$ and the impulse response $h(t)$ form the Fourier transform pair

$$h(t) \Leftrightarrow H(\omega)$$

and

$$f(t)*h(t) = g(t) \Leftrightarrow G(\omega) = F(\omega) \cdot H(\omega)$$

8.10 Exercises

1. Prove that

$$\int_{-\infty}^{\infty} u_0(t)\delta(t)dt = 1/2$$

2. Compute

$$\mathcal{F}\{te^{-at}u_0(t)\} \quad a > 0$$

3. Sketch the time and frequency waveforms of

$$f(t) = \cos\omega_0 t[u_0(t+T) - u_0(t-T)]$$

4. Derive the Fourier transform of

$$f(t) = A[u_0(t+3T) - u_0(t+T) + u_0(t-T) - u_0(t-3T)]$$

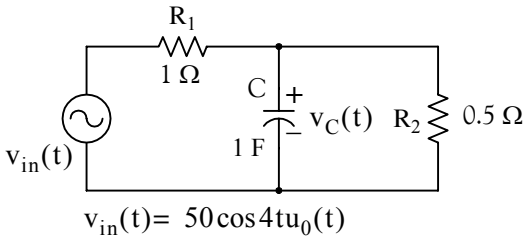
5. Derive the Fourier transform of

$$f(t) = \frac{A}{T}t[u_0(t+T) - u_0(t-T)]$$

6. Derive the Fourier transform of

$$f(t) = \left(\frac{A}{T}t + A\right)[u_0(t+T) - u_0(t)] + \left(-\frac{A}{T}t + A\right)\left[u_0(t) - u_0\left(t - \frac{T}{2}\right)\right]$$

7. For the circuit below, use the Fourier transform method to compute $v_C(t)$.



8. The input–output relationship in a certain network is

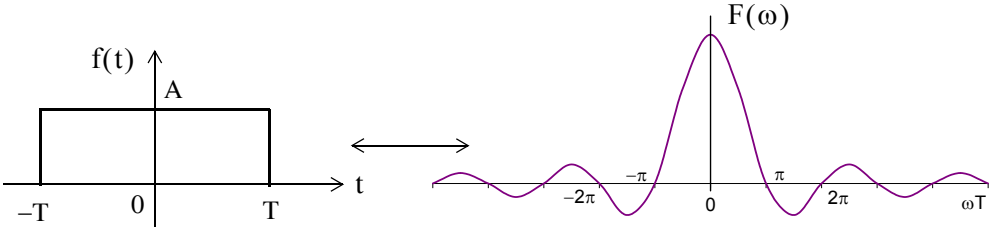
$$\frac{d^2}{dt^2}v_{out}(t) + 5\frac{d}{dt}v_{out}(t) + 6v_{out}(t) = 10v_{in}(t)$$

Use the Fourier transform method to compute $v_{out}(t)$ given that $v_{in}(t) = 2e^{-t}u_0(t)$.

9. In a bandpass filter, the lower and upper cutoff frequencies are $f_1 = 2$ Hz, and $f_2 = 6$ Hz respectively. Compute the 1 Ω energy of the input, and the percentage that appears at the output, if the input signal is $v_{in}(t) = 3e^{-2t}u_0(t)$ volts.

10. In Subsection 8.6.1, Page 8–27, we derived the Fourier transform pair

$$A[u_0(t+T)-u_0(t-T)] \Leftrightarrow 2AT\,\frac{\sin\omega T}{\omega T}$$



Compute the percentage of the 1 Ω energy of $f(t)$ contained in the interval $-\pi/T \le \omega \le \pi/T$ of $F(\omega)$.

11. In Subsection 8.6.2, Page 8-28, we derived the Fourier transform pair

$$A[u_0(t)-u_0(t-2T)] \Leftrightarrow 2ATe^{-j\omega T}\left(\frac{\sin\omega T}{\omega T}\right)$$

Use the fplot MATLAB command to plot the Fourier transform of this rectangular waveform.

8.11 Solutions to End–of–Chapter Exercises

1.

$$\int_{-\infty}^{\infty} u_0(t)\delta(t)dt = \int_{-\infty}^{\infty} u_0(t)\frac{d}{dt}u_0(t)dt = \int_{-\infty}^{\infty} u_0(t)d(u_0(t))$$

and since at $t = +\infty$, $u_0(t) = 1$ whereas at $t = -\infty$, $u_0(t) = 0$, we replace the limits of integration with 1 and 0. Then,

$$\int_0^1 u_0(t)d(u_0(t)) = \left.\frac{u_0^2(t)}{2}\right|_0^1 = 1/2$$

2.

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt = \int_0^{\infty} te^{-at}e^{-j\omega t}dt = \int_0^{\infty} te^{-(j\omega + a)t}dt$$

From tables of integrals

$$\int xe^{ax}dx = \frac{e^{ax}}{a^2}(ax - 1)$$

Then,

$$F(\omega) = \left.\frac{e^{-(j\omega + a)t}\cdot[-(j\omega + a)t - 1]}{(j\omega + a)^2}\right|_0^{\infty} = \left.\frac{[(j\omega + a)t + 1]}{e^{(j\omega + a)t}\cdot(j\omega + a)^2}\right|_{\infty}^{0}$$

With the upper limit of integration we obtain

$$F(\omega)\big|_{t = 0} = \frac{1}{(j\omega + a)^2}$$

To evaluate the lower limit of integration, we apply L'Hôpital's rule, i.e.,

$$\left.\frac{[(j\omega + a)t + 1]}{e^{(j\omega + a)t}\cdot(j\omega + a)^2}\right|_{\infty} = \lim_{t\to\infty}\frac{\frac{d}{dt}[(j\omega + a)t + 1]}{\frac{d}{dt}[e^{(j\omega + a)t}\cdot(j\omega + a)^2]} = \lim_{t\to\infty}\frac{(j\omega + a)}{(j\omega + a)e^{(j\omega + a)t}\cdot(j\omega + a)^2} = 0$$

and thus

$$F(\omega) = \frac{1}{(j\omega + a)^2}$$

Check:

$$F(\omega) = F(s)\big|_{s = j\omega}$$

and since

$$te^{-at}u_0(t) \Leftrightarrow \frac{1}{(s+a)^2}$$

it follows that

$$F(\omega) = \frac{1}{(s+a)^2}\bigg|_{s=j\omega} = \frac{1}{(j\omega+a)^2}$$

3.

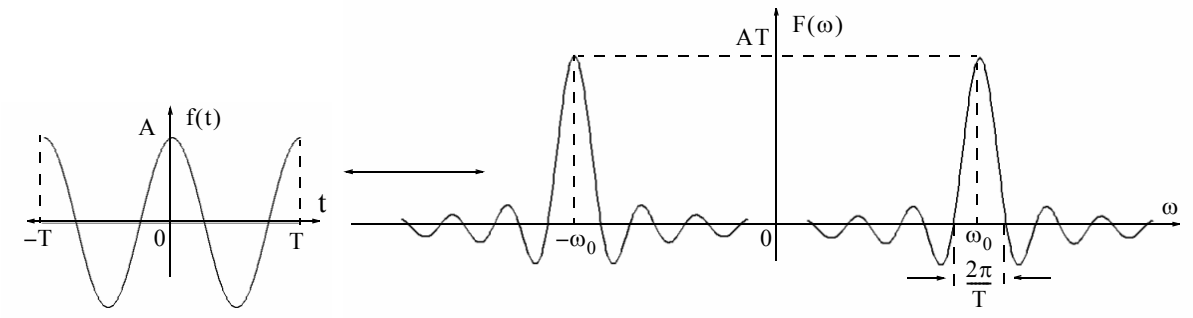
From Subsection 8.6.4, Page 8–30,

$$A\cos\omega_0 t[u_0(t+T)-u_0(t-T)] \Leftrightarrow AT\left[\frac{\sin[(\omega-\omega_0)T]}{(\omega-\omega_0)T}+\frac{\sin[(\omega+\omega_0)T]}{(\omega+\omega_0)T}\right]$$

and using the MATLAB script below,

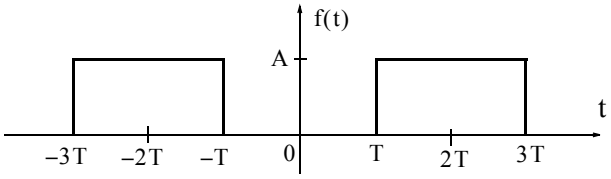
```
fplot('cos(x)',[–2*pi  2*pi  –1.2  1.2])
fplot('sin(x)./x',[–20  20  –0.4  1.2])
```

we obtain the plots below.



4.

$$f(t) = A[u_0(t+3T)-u_0(t+T)+u_0(t-T)-u_0(t-3T)]$$



From Subsection 8.6.1, Page 8–27,

$$A[u_0(t+T)-u_0(t-T)] \Leftrightarrow 2AT\,\frac{\sin\omega T}{\omega T}$$

and from the time shifting property,

$$f(t-t_0) \Leftrightarrow F(\omega)e^{-j\omega t_0}$$

Then,

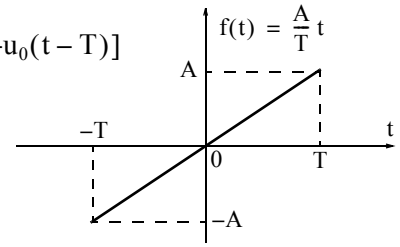
$$f(t) = A[u_0(t+3T) - u_0(t+T) + u_0(t-T) - u_0(t-3T)] \Leftrightarrow F(\omega) = 2AT\,\frac{\sin\omega T}{\omega T}\cdot(e^{j2\omega t} + e^{-j2\omega t})$$

or

$$F(\omega) = 4AT\,\frac{\sin\omega T}{\omega T}\cdot\left(\frac{e^{j2\omega t} + e^{-j2\omega t}}{2}\right) = 4AT\cos 2\omega T\,\frac{\sin\omega T}{\omega T}$$

5.

$$f(t) = \frac{A}{T}t[u_0(t+T) - u_0(t-T)]$$



$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt = \int_{-T}^{T}\frac{A}{T}te^{-j\omega t}dt = \frac{A}{T}\int_{-T}^{T}te^{-j\omega t}dt$$

From tables of integrals or integration by parts,

$$\int xe^{ax}dx = \frac{e^{ax}}{a^2}(ax-1)$$

Then,

$$F(\omega) = \frac{A}{T}\cdot\frac{e^{-j\omega t}}{(j\omega)^2}\cdot(-j\omega t - 1)\Big|_{-T}^{T} = \frac{A}{T}\cdot\frac{e^{-j\omega t}}{-\omega^2}\cdot(-j\omega t - 1)\Big|_{-T}^{T} = \frac{A}{T}\cdot\frac{e^{-j\omega t}}{-\omega^2}\cdot[-(j\omega t + 1)]\Big|_{-T}^{T}$$

$$= \frac{A}{\omega^2 T}[e^{-j\omega T}\cdot(j\omega T + 1) - e^{j\omega T}\cdot(-j\omega T + 1)]$$

$$= \frac{A}{\omega^2 T}(j\omega T\cdot e^{-j\omega T} + e^{-j\omega T} + j\omega T\cdot e^{j\omega T} - e^{j\omega T})$$

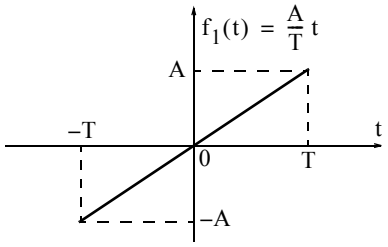
$$= \frac{A}{\omega^2 T}[j\omega T\cdot(e^{j\omega T} + e^{-j\omega T}) - (e^{j\omega T} - e^{-j\omega T})]$$

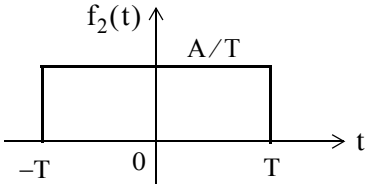
and multiplying both the numerator and denominator by $j2$ we obtain

$$F(\omega) = \frac{j2A}{\omega^2 T}\left[\frac{j\omega T(e^{j\omega T} + e^{-j\omega T})}{j2} - \frac{(e^{j\omega T} - e^{-j\omega T})}{j2}\right] = \frac{j2A}{\omega^2 T}(\omega T\cos\omega T - \sin\omega T)$$

We observe that since $f(t)$ is real and odd, $F(\omega)$ is imaginary and odd.

Alternate Solution:





The waveform of $f_2(t)$ is the derivative of the waveform of $f_1(t)$ and thus $f_1(t) = tf_2(t)$. From Subsection 8.6.1, Page 8–27,

$$f_2(t) \Leftrightarrow 2\frac{A}{T}T\,\frac{\sin\omega T}{\omega T} = 2A\,\frac{\sin\omega T}{\omega T}$$

From the frequency differentiation property, transform pair (8.48), Page 8–13,

$$(-jt)^n f(t) \Leftrightarrow \frac{d^n}{d\omega^n}F(\omega)$$

or

$$t^n f(t) \Leftrightarrow j^n\frac{d^n}{d\omega^n}F(\omega)$$

Then,

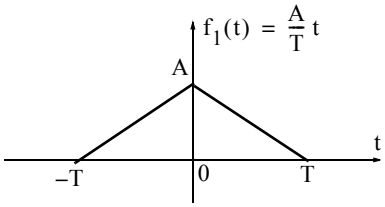
$$\mathcal{F}\{f_1(t) = tf_2(t)\} = \left(j\frac{d}{d\omega}\left(2A\,\frac{\sin\omega T}{\omega T}\right) = j2A\frac{d}{d\omega}\left(\frac{\sin\omega T}{\omega T}\right)\right)$$

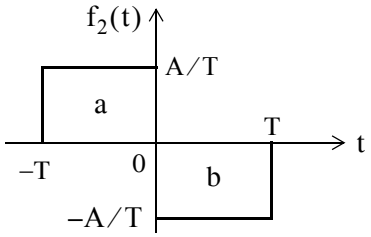
$$= j2A\left[\frac{(\omega T)T\cos\omega T - T(\sin\omega T)}{(\omega T)^2}\right] = \frac{j2A}{\omega^2 T}(\omega T\cos\omega T - \sin\omega T)$$

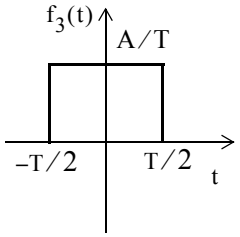
6.

We denote the given waveform as $f_1(t)$, that is,

$$f_1(t) = \left(\frac{A}{T}t + A\right)[u_0(t+T) - u_0(t)] + \left(-\frac{A}{T}t + A\right)\left[u_0(t) - u_0\left(t - \frac{T}{2}\right)\right]$$







We observe that $f_1(t)$ is the integral of $f_2(t)$. Therefore, we will find $F_2(\omega)$, and by integration we will find $F_1(\omega)$.

We begin by finding the Fourier Transform of the pulse denoted as $f_3(t)$, and using $F_3(\omega)$ and the time shifting and linearity properties, we will find $F_2(\omega)$.

$$F_3(\omega) = \int_{-T/2}^{T/2} \frac{A}{T} e^{-j\omega t} dt = \frac{A}{T}\left(\frac{e^{-j\omega t}}{-j\omega}\right)\Bigg|_{-T/2}^{T/2} = \frac{A}{T}\left(\frac{e^{-j\omega t}}{j\omega}\right)\Bigg|_{T/2}^{-T/2}$$

$$= \frac{A}{T}\left(\frac{e^{j\omega(T/2)} - e^{-j\omega(T/2)}}{j\omega}\right) = \frac{2A}{\omega T}\sin\frac{\omega T}{2} = A\frac{\sin(\omega T/2)}{\omega T/2}$$

Using the time shifting property

$$f(t - t_0) \Leftrightarrow F(\omega)e^{-j\omega t_0}$$

the Fourier transform of the left pulse a of $f_2(t)$ is

$$F_{2a}(\omega) = A\frac{\sin(\omega T/2)}{\omega T/2} \cdot e^{j\omega(T/2)}$$

Likewise, the Fourier transform of the right pulse b of $f_2(t)$ is

$$F_{2b}(\omega) = -A\frac{\sin(\omega T/2)}{\omega T/2} \cdot e^{-j\omega(T/2)}$$

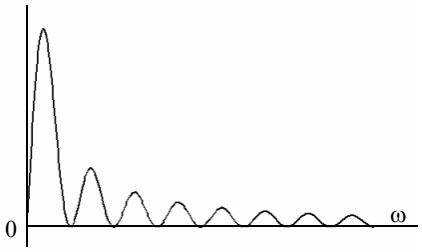
and using the linearity property we obtain

$$F_2(\omega) = F_{2a}(\omega) + F_{2b}(\omega) = A\frac{\sin(\omega T/2)}{\omega T/2} \cdot (e^{j\omega(T/2)} - e^{-j\omega(T/2)})$$

$$= j2A\frac{\sin(\omega T/2)}{\omega T/2} \cdot \left(\frac{e^{j\omega(T/2)} - e^{-j\omega(T/2)}}{j2}\right) = j2A\frac{\sin^2(\omega T/2)}{\omega T/2}$$

This $\sin^2(x)/x$ curve is shown below and it is created with the following MATLAB script:

```
fplot('sin(x./2).^2./x',[0 16*pi 0 0.5])
```



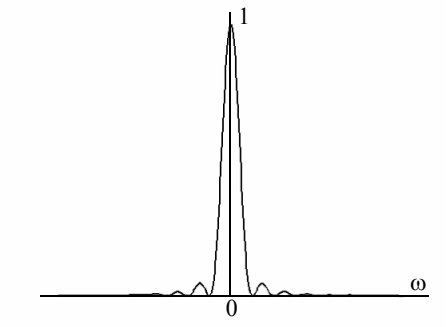
Now, we find $F_1(\omega)$ of the triangular waveform of $f_1(t)$ with the use of the integration property by multiplying $F_2(\omega)$ by $1/j\omega$. Thus,

$$F_1(\omega) = (1/j\omega)\cdot F_2(\omega) = \frac{1}{j\omega}\cdot j2A\frac{\sin^2(\omega T/2)}{\omega T/2} = \frac{2A}{\omega}\cdot\frac{\sin^2(\omega T/2)}{\omega T/2} = \frac{2A}{\omega^2 T/2}\cdot\sin^2(\omega T/2)$$

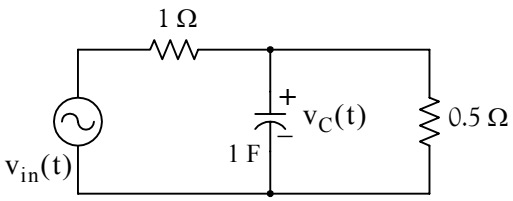
We can plot $F_1(\omega)$ by letting $T/2 = 1$. Then, $F_1(\omega)$ simplifies to the form $K[(\sin x)/x]^2$

This curve is shown below and it is created with the following MATLAB script.

```
fplot('(sin(x)./x).^2',[-8*pi 8*pi 0 1])
```



7.



By KCL

$$\frac{v_C(t)-v_{in}(t)}{1}+1\cdot\frac{dv_C}{dt}+\frac{v_C(t)}{0.5}=0$$

$$\frac{dv_C}{dt}+3v_C = v_{in}$$

Taking the Fourier transform of both sides we obtain

$$j\omega V_C(\omega)+3V_C(\omega) = V_{in}(\omega)$$

$$(j\omega+3)V_C(\omega) = V_{in}(\omega)$$

and since

$$V_{out}(\omega) = V_C(\omega),$$

$$(j\omega+3)V_{out}(\omega) = V_{in}(\omega)$$

and

$$H(\omega) = \frac{V_{out}(\omega)}{V_{in}(\omega)} = \frac{1}{j\omega+3}$$

where

$$v_{in}(t) = 50\cos 4t\,u_0(t) \Leftrightarrow V_{in}(\omega) = 50\pi[\delta(\omega-4)+\delta(\omega+4)]$$

Then,

$$V_{out}(\omega) = V_C(\omega) = H(\omega)\cdot V_{in}(\omega) = \frac{1}{j\omega+3}\cdot 50\pi[\delta(\omega-4)+\delta(\omega+4)]$$

and

$$v_C(t) = \mathcal{F}^{-1}\{V_C(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{50\pi[\delta(\omega-4)+\delta(\omega+4)]}{j\omega+3}\cdot e^{j\omega t}d\omega$$

$$= 25\int_{-\infty}^{\infty}\frac{\delta(\omega-4)}{j\omega+3}\cdot e^{j\omega t}d\omega + 25\int_{-\infty}^{\infty}\frac{\delta(\omega+4)}{j\omega+3}\cdot e^{j\omega t}d\omega$$

Next, using the sifting property of $\delta(\omega)$, we simplify the above to

$$v_C(t) = 25\left(\frac{e^{j4t}}{j4+3}+\frac{e^{-j4t}}{j4+3}\right) = 25\left(\frac{e^{j4t}}{5e^{j53.1°}}+\frac{e^{-j4t}}{5e^{-j53.1°}}\right) = 5(e^{j4t}\cdot e^{-j53.1°}+e^{-j4t}\cdot e^{j53.1°})$$

$$= 10\frac{e^{j(4t-53.1°)}+e^{-j(4t-53.1°)}}{2} = 10\cos(4t-53.1°)$$

8.

$$\frac{d^2}{dt^2}v_{out}(t)+5\frac{d}{dt}v_{out}(t)+6v_{out}(t) = 10v_{in}(t) = 2e^{-t}u_0(t)$$

Taking the Fourier transform of both sides we obtain

$$[(j\omega)^2+5j\omega+6]V_{out}(\omega) = 10V_{in}(\omega) = 10\cdot\frac{2}{j\omega+1}$$

$$[(j\omega+2)\cdot(j\omega+3)]V_{out}(\omega) = 10V_{in}(\omega) = \frac{20}{j\omega+1}$$

$$V_{out}(\omega) = \frac{20}{(j\omega+1)\cdot(j\omega+2)\cdot(j\omega+3)}$$

We use the following MATLAB script for partial fraction expansion where we let $j\omega = s$.

```
syms s; collect((s+1)*(s+2)*(s+3))
```

```
ans =
 s^3+6*s^2+11*s+6
```

```
num=[0 0 0 20]; den=[1 6 11 6]; [num,den]=residue(num,den); fprintf(' \n');...
fprintf('r1 = %4.2f \t', num(1)); fprintf('p1 = %4.2f', den(1)); fprintf(' \n');...
fprintf('r2 = %4.2f \t', num(2)); fprintf('p2 = %4.2f', den(2)); fprintf(' \n');...
fprintf('r3 = %4.2f \t', num(3)); fprintf('p3 = %4.2f', den(3))
```

```
r1 =  10.00   p1 = -3.00
r2 = -20.00   p2 = -2.00
r3 =  10.00   p3 = -1.00
```

Then,

$$V_{out}(\omega) = \frac{20}{(j\omega+1)\cdot(j\omega+2)\cdot(j\omega+3)} + \frac{10}{(j\omega+1)} + \frac{-20}{(j\omega+2)} + \frac{10}{(j\omega+3)}$$

and thus

$$v_{out}(t) = \mathcal{F}^{-1}\{V_{out}(\omega)\} = 10e^{-t} - 20e^{-2t} + 10e^{-3t}$$
$$= 10(e^{-t} - 2e^{-2t} + e^{-3t})u_0(t)$$

9.

The input energy in joules is

$$W_{in} = \int_{-\infty}^{\infty} |v_{in}(t)|^2 dt = \int_0^{\infty} |3e^{-2t}|^2 dt = \int_0^{\infty} |3e^{-2t}|^2 dt = \int_0^{\infty} 9e^{-4t} dt$$

$$= \frac{9e^{-4t}}{-4}\bigg|_0^{\infty} = \frac{9e^{-4t}}{4}\bigg|_{\infty}^{0} = \frac{9}{4} = 2.25 \text{ J}$$

and the Fourier transform $F_{in}(\omega)$ of the input $v_{in}(t)$ is

$$\mathcal{F}\{v_{in}(t)\} = \mathcal{F}\{3e^{-2t}u_0(t)\} = \frac{3}{j\omega+2}$$

The energy at the output for the frequency interval $2 \text{ Hz} \le f \le 6 \text{ Hz}$ or $4\pi \text{ rad} \le \omega \le 12\pi \text{ rad}$ is

$$W_{out} = \frac{1}{2\pi}\int_{-\infty}^{\infty} |F(\omega)|^2 d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} \left|\frac{3}{j\omega+2}\right|^2 d\omega = \frac{1}{2\pi}\int_{4\pi}^{12\pi} \frac{9}{\omega^2+2^2} d\omega$$

and from tables of integrals

$$\int \frac{1}{x^2+a^2} dx = \frac{1}{a}\text{atan}\frac{x}{a}$$

Then,

$$W_{out} = \frac{9}{2\pi}\cdot\frac{1}{2}\text{atan}\frac{\omega}{2}\Big|_{4\pi}^{12\pi} = \frac{9}{4\pi}\left(\text{atan}\frac{12\pi}{2} - \text{atan}\frac{4\pi}{2}\right) = \frac{9}{4\pi}(\text{atan}\,6\pi - \text{atan}\,2\pi)$$

```
fprintf(' \n'); fprintf('atan(6*pi) = %4.2f \t', atan(6*pi)); fprintf('atan(2*pi) = %4.2f', atan(2*pi))
```

```
atan(6*pi) = 1.52    atan(2*pi) = 1.41
```

and thus

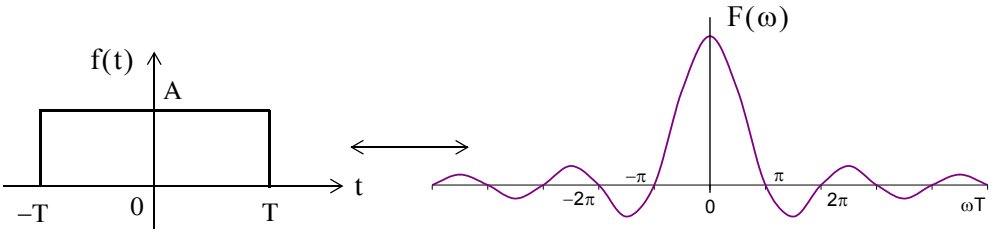
$$W_{out} = \frac{9}{4\pi}(1.52 - 1.41) = 0.08 \text{ J}$$

Therefore, the percentage of the input appearing at the output is

$$\frac{W_{out}}{W_{in}} \times 100 = \frac{0.08}{2.25} = 3.56\%$$

10.

$$A[u_0(t+T) - u_0(t-T)] \Leftrightarrow 2AT\,\frac{\sin\omega T}{\omega T}$$



First, we compute the total energy of the pulse in terms of $f(t)$.

$$W_{total} = \int_{-\infty}^{\infty}|f(t)|^2 dt = \int_{-T}^{T} A^2 dt$$

and since $f(t)$ is an even function,

$$W_{total} = 2\int_{0}^{T} A^2 dt = 2A^2 t\Big|_0^T = 2A^2 T$$

Next, we denote the energy in the frequency interval $-\pi/T \text{ rad} \le \omega \le \pi/T \text{ rad}$ as W_{out} in the frequency domain we obtain

$$W_{out} = \frac{1}{2\pi}\int_{-\infty}^{\infty}|F(\omega)|^2 d\omega = \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T}\left|2AT\,\frac{\sin\omega T}{\omega T}\right|^2 d\omega$$

and since $F(\omega)$ is an even function,

$$W_{out} = 2\frac{1}{2\pi}\int_0^{\pi/T} 4A^2T^2\,\frac{\sin^2\omega T}{(\omega T)^2}d\omega = \frac{4A^2T^2}{\pi}\int_0^{\pi/T}\frac{\sin^2\omega T}{(\omega T)^2}d\omega \quad (1)$$

For simplicity, we let $\omega T = y$. Then, $\omega = y/T$ and $d\omega = (1/T)dy$. Also, when $\omega = 0$, $y = 0$, and when $\omega = \pi/T$ or $\omega T = \pi$, $y = \pi$. With these substitutions we express (1) as

$$W_{out} = \frac{4A^2T^2}{\pi}\int_0^{\pi}\frac{\sin^2 y}{((y/T)T)^2}dy = \frac{4A^2T^2}{\pi T}\int_0^{\pi}\frac{\sin^2 y}{y^2}dy = \frac{4A^2T}{\pi}\int_0^{\pi}\frac{\sin^2 y}{y^2}dy \quad (2)$$

But the last integral in (2) is an improper integral and does not appear in tables of integrals.* Therefore, we will attempt to simplify (2) using integration by parts. We start with the familiar relation

$$d(uv) = udv + vdu$$

from which

$$\int d(uv) = \int udv + \int vdu$$

or

$$\int udv = uv - \int vdu$$

Letting $u = \sin^2 y$ and $dv = 1/y^2$, it follows that $du = 2\cos y\sin y = \sin 2y$ and $v = -1/y$. With these substitutions (2) is written as

$$\begin{aligned} W_{out} &= \frac{4A^2T}{\pi}\left[\frac{\sin^2 y}{-y}\bigg|_0^{\pi} - \int_0^{\pi}\frac{-1}{y}\sin 2y\,dy\right] = \frac{4A^2T}{\pi}\left[0 + \int_0^{\pi}\frac{\sin 2y}{y}dy\right] \\ &= 2\cdot\frac{4A^2T}{\pi}\int_0^{\pi}\frac{\sin 2y}{2y}dy = \frac{8A^2T}{\pi}\int_0^{\pi}\frac{\sin 2y}{2y}dy \end{aligned} \quad (3)$$

The last integral in (3) is also an improper integral. Fortunately, some handbooks of mathematical tables include numerical values of the integral

$$\int_0^{\pi}\frac{\sin x}{x}dx$$

* *It is shown in Advanced Calculus textbooks that if the upper limit is ∞, then*

$$\int_0^{\infty}\frac{\sin x}{x}dx = \int_0^{\infty}\frac{\sin^2 x}{x^2}dx = \frac{\pi}{2}$$

but for other finite limits are not equal.

for arguments of πx in the interval $0 \le x \le 10$. Then, replacing $2y$ with w we obtain $w = 2y$, $y = w/2$, $dy = (1/2)/dw$, and for $y = 0$, $w = 0$ whereas for $y = \pi$, $w = 2\pi$. Then, by substitution into (3) we obtain

$$W_{out} = \frac{8A^2T}{\pi}\int_0^{2\pi}\frac{\sin w}{w}\left(\frac{1}{2}dw\right) = \frac{4A^2T}{\pi}\int_0^{2\pi}\frac{\sin w}{w}dw \quad (4)$$

From Table 5.3 of Handbook of Mathematical Functions, 1972 Edition, Dover Publications, with $\pi x = 2\pi$ or $x = 2$ we obtain

$$\int_0^{2\pi}\frac{\sin w}{w}dw\Bigg|_{x=2} = 1.418$$

and thus (4) reduces to

$$W_{out} = \frac{4A^2T}{\pi}1.418$$

Therefore, the percentage of the output for the frequency interval $-\pi/T$ rad $\le \omega \le \pi/T$ rad is

$$\frac{W_{out}}{W_{total}} \times 100\% = \frac{(4A^2T/\pi)\cdot 1.418}{2A^2T} \times 100\% = \frac{2 \times 1.418}{\pi} \times 100\% = 90.3\%$$

Since this computation involves numerical integration, we can obtain the same result much faster and easier with MATLAB as follows:

First, we define the function **fourierxfm1** and we save it as an **.m** file as shown below. This file must be created with MATLAB's editor (or any other editor) and saved as an **.m** file in a drive that has been added to MATLAB's path.

```
function y1=fourierxfm1(x)
x=x+(x==0)*eps;% This statement avoids the sin(0)/0 value.
% It says that if x=0, then (x==0) = 1
% but if x is not zero, then (x==0) = 0
% and eps is approximately equal to 2.2e-16
% It is used to avoid division by zero.
y1=sin(x)./x;
```

Then, at MATLAB's Command prompt, we write and execute the program below.

```
% The quad function below performs numerical integration from 0 to 2*pi
% using a form of Simpson's rule of numerical integration.

value1=quad('fourierxfm1',0,2*pi)
```

```
value1 =
  1.4182
```

We could also have used numerical integration with the integral

$$\int_0^{\pi} \frac{\sin^2 x}{x^2} dx$$

thereby avoiding the integration by parts procedure. Shown below are the function fourierxfm2 which is saved as an .m file and the program execution using this function.

```
function y2=fourierxfm2(x)
x=x+(x==0)*eps;
y2=(sin(x)./x).^2;
```

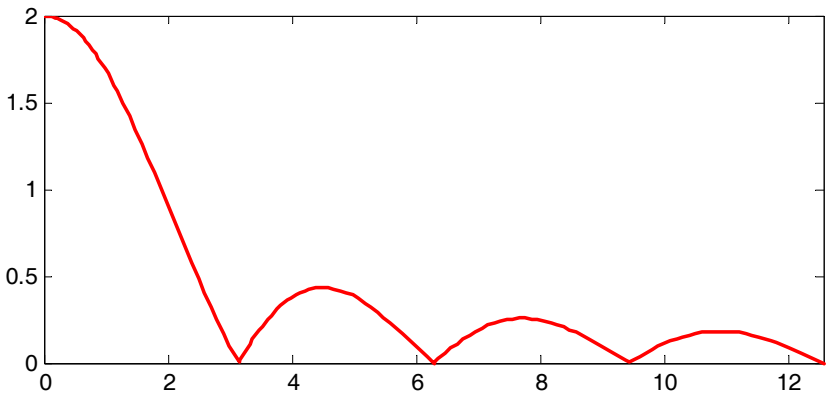
and after this file is saved, we execute the statement below observing that the limits of integration are from 0 to π.

```
value2=quad('fourierxfm2',0,pi)
```

```
value2 =
  1.4182
```

11.

```
fplot('abs(2.*exp(–j.*w)*(sin(w)/w))',[0 4*pi 0 2])
```



Chapter 9

Discrete–Time Systems and the $\mathcal{Z}$ Transform

This chapter is devoted to discrete–time systems, and introduces the one–sided $\mathcal{Z}$ Transform. The definition, theorems, and properties are discussed, and the $\mathcal{Z}$ transforms of the most common discrete–time functions are derived. The discrete transfer function is also defined, and several examples are given to illustrate its application. The Inverse $\mathcal{Z}$ transform, and the methods available for finding it, are also discussed.

9.1 Definition and Special Forms

The $\mathcal{Z}$ *transform* performs the transformation from the domain of discrete–time signals, to another domain which we call $z-\text{domain}$. It is used with discrete–time signals,* the same way the Laplace and Fourier transforms are used with continuous–time signals. The $\mathcal{Z}$ transform yields a frequency domain description for discrete–time signals, and forms the basis for the design of digital systems, such as digital filters. Like the Laplace transform, there is the one–sided, and the two–sided $\mathcal{Z}$ transform. We will restrict our discussion to the one–sided $\mathcal{Z}$ transform $F(z)$ of a discrete–time function $f[n]$ defined as

$$\boxed{F(z) = \sum_{n=0}^{\infty} f[n]z^{-n}} \tag{9.1}$$

and the *Inverse* $\mathcal{Z}$ *transform* is defined as

$$\boxed{f[n] = \frac{1}{j2\pi}\oint F(z)z^{k-1}dz} \tag{9.2}$$

We can obtain a discrete–time waveform from an analog (continuous or with a finite number of discontinuities) signal, by multiplying it by a train of impulses. We denote the continuous signal as $f(t)$, and the impulses as

$$\delta[n] = \sum_{n=0}^{\infty} \delta[t-nT] \tag{9.3}$$

Multiplication of $f(t)$ by $\delta[n]$ produces the signal $g(t)$ defined as

$$g(t) = f(t)\cdot\delta[n] = f(t)\sum_{n=0}^{\infty}\delta[t-nT] \tag{9.4}$$

* *Whereas continuous-time signals are described by differential equations, discrete–time signals are described by difference equations.*

These signals are shown in Figure 9.1.

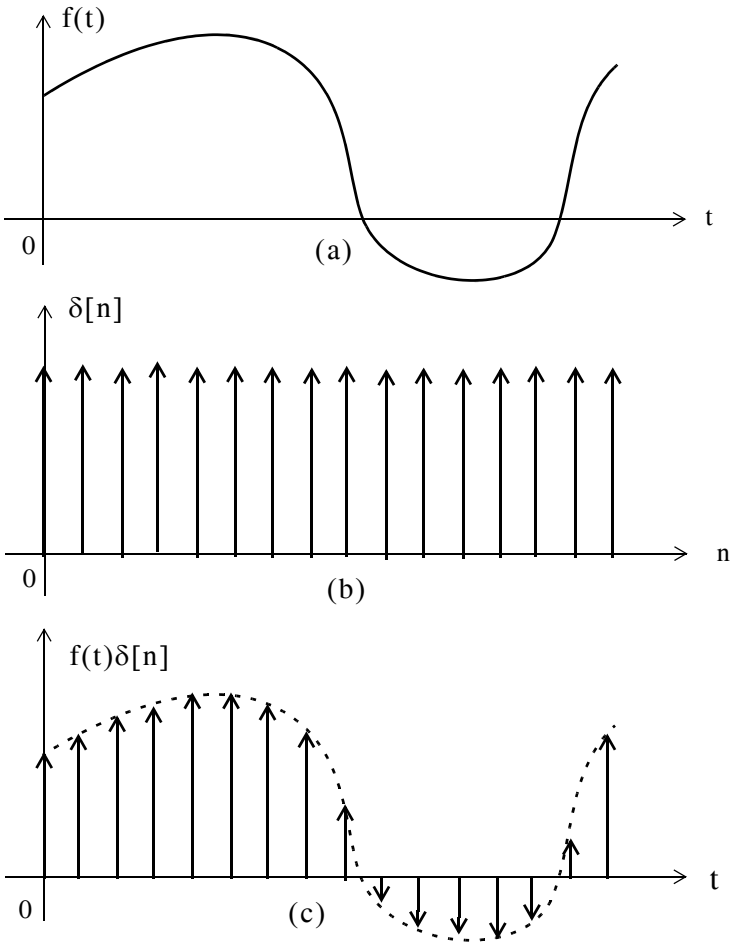


Figure 9.1. Formation of discrete–time signals

Of course, after multiplication by $\delta[n]$, the only values of $f(t)$ which are not zero, are those for which $t = nT$, and thus we can express (9.4) as

$$g(t) = f[nT]\sum_{n=0}^{\infty}\delta[t-nT] = \sum_{n=0}^{\infty} f[nT]\delta[t-nT] \tag{9.5}$$

Next, we recall from Chapter 2, that the $t-\text{domain}$ to $s-\text{domain}$ transform pairs for the delta function are $\delta(t) \Leftrightarrow 1$ and $\delta(t-T) \Leftrightarrow e^{-sT}$. Therefore, taking the Laplace transform of both sides of (9.5), and, for simplicity, letting $f[nT] = f[n]$, we obtain

$$G(s) = \mathscr{L}\left\{ f[n]\sum_{n=0}^{\infty}\delta[t-nT] \right\} = f[n]\sum_{n=0}^{\infty} e^{-nsT} = \sum_{n=0}^{\infty} f[n]e^{-nsT} \tag{9.6}$$

Relation (9.6), with the substitution $z = e^{sT}$, becomes the same as (9.1), and like s, z is also a complex variable.

The $\mathcal{Z}$ and Inverse $\mathcal{Z}$ transforms are denoted as

$$\boxed{F(z) = \mathcal{Z}\{f[n]\}} \tag{9.7}$$

and

$$\boxed{f[n] = \mathcal{Z}^{-1}\{F(z)\}} \tag{9.8}$$

The function $F(z)$, as defined in (9.1), is a series of complex numbers and converges outside the circle of radius R, that is, it converges (approaches a limit) when $|z| > R$. In complex variables theory, the radius R is known as the *radius of absolute convergence*.

In the complex $z-plane$ the *region of convergence* is the set of z for which the magnitude of $F(z)$ is finite, and the *region of divergence* is the set of z for which the magnitude of $F(z)$ is infinite.

9.2 Properties and Theorems of the $\mathcal{Z}$ Transform

The properties and theorems of the $\mathcal{Z}$ transform are similar to those of the Laplace transform. In this section, we will state and prove the most common $\mathcal{Z}$ transforms listed in Subsections 9.2.1 through 9.2.12 below.

9.2.1 Linearity

$$\boxed{af_1[n] + bf_2[n] + cf_3[n] + \ldots \Leftrightarrow aF_1(z) + bF_2(z) + cF_3(z) + \ldots} \tag{9.9}$$

where $a, b, c, \ldots$ are arbitrary real or complex constants.

Proof:

The proof is easily obtained by application of the definition of the $\mathcal{Z}$ transform to each term on the left side.

In our subsequent discussion, we will denote the *discrete unit step function* as $u_0[n]$.

9.2.2 Shift of $f[n]u_0[n]$ in the Discrete–Time Domain

$$\boxed{f[n-m]u_0[n-m] \Leftrightarrow z^{-m}F(z)} \tag{9.10}$$

Proof:

Applying the definition of the $\mathcal{Z}$ transform, we obtain

$$\mathcal{Z}\{f[n-m]u_0[n-m]\} = \sum_{n=0}^{\infty} f[n-m]u_0[n-m]z^{-n}$$

and since $u_0[n-m] = 0$ for $n < m$ and $u_0[n-m] = 1$ for $n > m$, the above expression reduces to

$$\mathcal{Z}\{f[n-m]\} = \sum_{n=0}^{\infty} f[n-m]z^{-n}$$

Now, we let $n-m = k$; then, $n = k+m$, and when $n-m = 0$ or $n = m$, $k = 0$. Therefore,

$$\mathcal{Z}\{f[n-m]\} = \sum_{k=0}^{\infty} f[k]z^{-(k+m)} = \sum_{k=0}^{\infty} f[k]z^{-k}z^{-m} = z^{-m}\sum_{k=0}^{\infty} f[k]z^{-k} = z^{-m}F(z)$$

9.2.3 Right Shift in the Discrete–Time Domain

This property is a generalization of the previous property, and allows use of non–zero values for $n < 0$. The transform pair is

$$\boxed{f[n-m] \Leftrightarrow z^{-m}F(z) + \sum_{n=0}^{m-1} f[n-m]z^{-n}} \qquad (9.11)$$

Proof:

By application of the definition of the $\mathcal{Z}$ transform, we obtain

$$\mathcal{Z}\{f[n-m]\} = \sum_{n=0}^{\infty} f[n-m]z^{-n}$$

We let $n-m = k$; then, $n = k+m$, and when $n = 0$, $k = -m$. Therefore,

$$\mathcal{Z}\{f[n-m]\} = \sum_{k=-m}^{\infty} f[k]z^{-(k+m)} = \sum_{k=-m}^{\infty} f[k]z^{-k}z^{-m} = z^{-m}\sum_{k=-m}^{\infty} f[k]z^{-k}$$

$$= z^{-m}\left[\sum_{k=-m}^{-1} f[k]z^{-k} + \sum_{k=0}^{\infty} f[k]z^{-k}\right] = z^{-m}\left[F(z) + \sum_{k=-m}^{-1} f[k]z^{-k}\right]$$

When $k = -m$, $n = 0$, and when $k = -1$, $n = m-1$. Then, by substitution into the last summation term above, we obtain

$$\mathcal{Z}\{f[n-m]\} = z^{-m}\left[F(z) + \sum_{n=0}^{m-1} f[n-m]z^{m-n}\right] = z^{-m}F(z) + \sum_{n=0}^{m-1} f[n-m]z^{-n}$$

and this is the same as (9.11).

For $m = 1$, (9.11) reduces to

$$\boxed{f[n-1] \Leftrightarrow z^{-1}F(z) + f[-1]} \tag{9.12}$$

and for $m = 2$, reduces to

$$\boxed{f[n-2] \Leftrightarrow z^{-2}F(z) + f[-2] + z^{-1}f[-1]} \tag{9.13}$$

9.2.4 Left Shift in the Discrete–Time Domain

$$\boxed{f[n+m] \Leftrightarrow z^{m}F(z) + \sum_{n=-m}^{-1} f[n+m]z^{-n}} \tag{9.14}$$

that is, if $f[n]$ is a discrete–time signal, and m is a positive integer, the mth left shift of $f[n]$ is $f[n+m]$.

Proof:

$$\mathcal{Z}\{f[n+m]\} = \sum_{n=0}^{\infty} f[n+m]z^{-n}$$

We let $n+m = k$; then, $n = k-m$, and when $n = 0$, $k = m$. Then,

$$\mathcal{Z}\{f[n+m]\} = \sum_{k=m}^{\infty} f[k]z^{-(k-m)} = \sum_{k=m}^{\infty} f[k]z^{-k}z^{m}$$

$$= z^{m}\left[\sum_{k=0}^{\infty} f[k]z^{-k} - \sum_{k=0}^{m-1} f[k]z^{-k}\right] = z^{m}\left[F(z) - \sum_{k=0}^{m-1} f[k]z^{k}\right.$$

When $k = 0$, $n = -m$, and when $k = m-1$, $n = -1$. Then, by substitution into the last summation term of the above expression, we obtain

$$\mathcal{Z}\{f[n+m]\} = z^{m}\left[F(z) + \sum_{n=-m}^{-1} f[n+m]z^{-(n+m)}\right] = z^{m}F(z) + \sum_{n=-m}^{-1} f[n+m]z^{-n}$$

and this is the same as (9.14).

For $m = 1$, the above expression reduces to

$$\boxed{\mathcal{Z}\{f[n+1]\} = zF(z) - f[0]z} \tag{9.15}$$

and for $m = 2$, reduces to

$$\boxed{\mathcal{Z}\{f[n+2]\} = z^2F(z) - f[0]z^2 - f[1]z} \tag{9.16}$$

9.2.5 Multiplication by a^n in the Discrete–Time Domain

$$\boxed{a^n f[n] \Leftrightarrow F\left(\frac{z}{a}\right)} \tag{9.17}$$

Proof:

$$\mathcal{Z}\{a^n f[n]\} = \sum_{k=0}^{\infty} a^n f[n]z^{-n} = \sum_{k=0}^{\infty} \frac{1}{a^{-n}} f[n]z^{-n} = \sum_{k=0}^{\infty} f[n]\left(\frac{z}{a}\right)^{-n} = F\left(\frac{z}{a}\right)$$

9.2.6 Multiplication by e^{-naT} in the Discrete–Time Domain

$$\boxed{e^{-naT} f[n] \Leftrightarrow F(e^{aT}z)} \tag{9.18}$$

Proof:

$$\mathcal{Z}\{e^{-naT} f[n]\} = \sum_{k=0}^{\infty} e^{-naT} f[n]z^{-n} = \sum_{k=0}^{\infty} f[n](e^{aT}z)^{-n} = F(e^{aT}z)$$

9.2.7 Multiplication by n and n^2 in the Discrete–Time Domain

$$\boxed{\begin{aligned} nf[n] &\Leftrightarrow -z\frac{d}{dz}F(z) \\ n^2 f[n] &\Leftrightarrow z\frac{d}{dz}F(z) + z^2\frac{d^2}{dz^2}F(z) \end{aligned}} \tag{9.19}$$

Proof:

By definition,

$$F(z) = \sum_{n=0}^{\infty} f[n]z^{-n}$$

and taking the first derivative of both sides with respect to z, we obtain

$$\frac{d}{dz}F(z) = \sum_{n=0}^{\infty} (-n)f[n]z^{-n-1} = -z^{-1}\sum_{n=0}^{\infty} nf[n]z^{-n}$$

Multiplication of both sides by $-z$ yields

$$\sum_{n=0}^{\infty} nf[n]z^{-n} = -z\frac{d}{dz}F(z)$$

Differentiating one more time, we obtain the second pair in (9.19).

9.2.8 Summation in the Discrete–Time Domain

$$\boxed{\sum_{m=0}^{n} f[m] \Leftrightarrow \left(\frac{z}{z-1}\right)F(z)} \tag{9.20}$$

that is, the $\mathcal{Z}$ transform of the sum of the values of a signal, is equal to $z/(z-1)$ times the $\mathcal{Z}$ transform of the signal. This property is equivalent to time integration in the continuous time domain since integration in the discrete–time domain is summation. We will see on the next section that the term $z/(z-1)$ is the $\mathcal{Z}$ transform of the discrete unit step function $u_0[n]$, and recalling that in the $s-$domain

$$u_0(t) \Leftrightarrow \frac{1}{s}$$

and

$$\int_0^t f(\tau)d\tau \Leftrightarrow \frac{F(s)}{s}$$

then, the similarity of the Laplace and $\mathcal{Z}$ transforms becomes apparent.

Proof:

Let

$$y[n] = \sum_{m=0}^{n} x[m] \tag{9.21}$$

and let us express (9.21) as

$$y[n] = \sum_{m=0}^{n-1} x[m] + x[n] \tag{9.22}$$

Since the summation symbol in (9.21) is $y[n]$, then the summation symbol in (9.22) is $y[n-1]$, and thus we can write (9.22) as

$$y[n] = y[n-1] + x[n] \tag{9.23}$$

Next, we take the $\mathcal{Z}$ transform of both sides of (9.23), and using the property

$$x[n-m]u_0[n-m] \Leftrightarrow z^{-m}X(z)$$

we obtain

$$Y(z) = z^{-1}Y(z) + X(z)$$

or

$$Y(z) = \frac{1}{1 - z^{-1}}\,X(z) = \frac{z}{z - 1}\,X(z)$$

and this relation is the same as (9.20).

9.2.9 Convolution in the Discrete–Time Domain

Let the impulse response of a discrete–time system be denoted as $h[n]$, that is, an impulse $\delta[n]$, produces a response $h[n]$. Likewise, a delayed impulse $\delta[n-m]$ produces a delayed response $h[n-m]$, and so on. Therefore, any discrete–time input signal can be considered as an impulse train, in which each impulse has a weight equal to its corresponding sampled value. Then, for any other input $x[0], x[1], x[2], \ldots, x[m]$, we obtain

$$\begin{gathered} x[0]\delta[0] \rightarrow x[0]h[n] \\ x[1]\delta[n-1] \rightarrow x[1]h[n-1] \\ x[2]\delta[n-2] \rightarrow x[2]h[n-2] \\ \cdots \\ x[m]\delta[n-m] \rightarrow x[m]h[n-m] \end{gathered}$$

and the response at any arbitrary value *m*, is obtained by summing all the components that have occurred up to that point, that is, if $y[n]$ is the output due to the input $x[m]$ convolved with $h[n]$, then,

$$y[n] = \sum_{m=0}^{n} x[m]h[n-m] \tag{9.24}$$

or

$$y[n] = \sum_{m=0}^{n} h[n-m]x[m] \tag{9.25}$$

We will now prove that convolution in the discrete–time domain corresponds to multiplication in the $\mathcal{Z}$ domain, that is,

$$\boxed{f_1[n] * f_2[n] \Leftrightarrow F_1(z) \cdot F_2(z)} \tag{9.26}$$

Proof:

Taking the $\mathcal{Z}$ transform of both sides of (9.24), we obtain

$$Y(z) = \mathcal{Z}\left\{\sum_{m=0}^{\infty} x[m]h[n-m]\right\} = \sum_{n=0}^{\infty}\left[\sum_{m=0}^{\infty} x[m]h[n-m]\right]z^{-n}$$

and interchanging the order of the summation, we obtain

$$Y(z) = \sum_{m=0}^{\infty}\left[\sum_{n=0}^{\infty} x[m]h[n-m]z^{-n}\right] = \sum_{m=0}^{\infty} x[m]\sum_{n=0}^{\infty} h[n-m]\,z^{-n}$$

Next, we let $k = n-m$, then, $n = k+m$, and thus,

$$Y(z) = \sum_{m=0}^{\infty} x[m]\sum_{n=0}^{\infty} h[k]z^{-(k+m)} = \sum_{m=0}^{\infty} x[m]z^{-m}\sum_{n=0}^{\infty} h[k]z^{-k}$$

or

$$Y(z) = X(z)\cdot H(z) \tag{9.27}$$

9.2.10 Convolution in the Discrete–Frequency Domain

If $f_1[n]$ and $f_2[n]$ are two sequences with $\mathcal{Z}$ transforms $F_1(z)$ and $F_2(z)$ respectively, then,

$$f_1[n]\cdot f_2[n] \Leftrightarrow \frac{1}{j2\pi}\oint xF_1(v)F_2\left(\frac{z}{v}\right)v^{-1}dv \tag{9.28}$$

where v is a dummy variable, and $\oint$ is a closed contour inside the overlap convergence regions for $X_1(v)$ and $X_2(z/v)$. The proof requires contour integration; it will not be provided here.

9.2.11 Initial Value Theorem

$$\boxed{f[0] = \lim_{z\to\infty} X(z)} \tag{9.29}$$

Proof:

For all $n \geq 1$, as $z \to \infty$

$$z^{-n} = \frac{1}{z^n} \to 0$$

and under these conditions $f[n]z^{-n} \to 0$ also. Taking the limit as $z \to \infty$ in

$$F(z) = \sum_{n=0}^{\infty} f[n]z^{-n}$$

we observe that the only non–zero value in the summation is that of $n = 0$. Then,

$$\sum_{n=0}^{\infty} f[n]z^{-n} = f[0]\,z^{-0} = f[0]$$

Therefore,

$$\lim_{z\to\infty} X(z) = f[0]$$

9.2.12 Final Value Theorem

This theorem states that if $f[n]$ approaches a limit as $n \to \infty$, we can find that limit, if it exists, by multiplying the $\mathcal{Z}$ transform of $f[n]$ by $(z-1)$, and taking the limit of the product as $z \to 1$. That is,

$$\boxed{\lim_{n\to\infty} f[n] = \lim_{z\to 1}(z-1)F(z)} \tag{9.30}$$

Proof:

Let us consider the $\mathcal{Z}$ transform of the sequence $f[n+1]-f[n]$, i.e.,

$$\mathcal{Z}\{f[n+1]-f[n]\} = \sum_{n=0}^{\infty}(f[n+1]-f[n])z^{-n}$$

We replace the upper limit of the summation with *k*, and we let $k \to \infty$. Then,

$$\mathcal{Z}\{f[n+1]-f[n]\} = \lim_{k\to\infty}\left[\sum_{n=0}^{k}(f[n+1]-f[n])z^{-n}\right] \tag{9.31}$$

From (9.15),

$$\mathcal{Z}\{f[n+1]\} = zF(z)-f[0]z \tag{9.32}$$

and by substitution of (9.32) into (9.31), we obtain

$$zF(z)-f[0]z-F(z) = \lim_{k\to\infty}\left[\sum_{n=0}^{k}(f[n+1]-f[n])z^{-n}\right]$$

Taking the limit as $z \to 1$ on both sides, we obtain

$$\lim_{z\to 1}\{(z-1)F(z)-f[0]z\} = \lim_{z\to 1}\left\{\lim_{k\to\infty}\left[\sum_{n=0}^{k}(f[n+1]-f[n])z^{-n}\right]\right\}$$

$$= \lim_{k\to\infty}\left[\sum_{n=0}^{k}\lim_{z\to 1}(f[n+1]-f[n])z^{-n}\right]$$

$$\lim_{z\to 1}(z-1)F(z)-\lim_{z\to 1}f[0]z=\lim_{k\to\infty}\left[\sum_{n=0}^{k}\lim_{z\to 1}\{f[n+1]-f[n]\}z^{-n}\right]$$

$$\lim_{z\to 1}(z-1)F(z)-f[0]=\lim_{k\to\infty}\left[\sum_{n=0}^{k}\{f[n+1]\}-f[n]\right]$$

$$=\lim_{k\to\infty}\{f[k]-f[0]\}=\lim_{k\to\infty}f[k]-f[0]$$

$$\lim_{k\to\infty}f[k]=\lim_{z\to 1}(z-1)F(z)$$

We must remember, however, that if the sequence $f[n]$ does not approach a limit, the final value theorem is invalid. The right side of (9.30) may exist even though $f[n]$ does not approach a limit. In instances where we cannot determine whether $f[n]$ exists or not, we can be certain that it exists if $X(z)$ can be expressed in a proper rational form as

$$X(z)=\frac{A(z)}{B(z)}$$

where $A(z)$ and $B(z)$ are polynomials with real coefficients.

For convenience, we summarize the properties and theorems of the $\mathcal{Z}$ transform in Table 9.1.

9.3 The $\mathcal{Z}$ Transform of Common Discrete–Time Functions

In this section we will provide several examples to find the $\mathcal{Z}$ transform of some discrete–time functions. In this section, we will derive the $\mathcal{Z}$ transforms of the most common discrete–time functions in Subsections 9.3.1 through 9.3.5 below.

9.3.1 The Transform of the Geometric Sequence

The *geometric sequence* is defined as

$$f[n]=\begin{cases}0 & n=-1,-2,-3,\ldots\\ a^n & n=0,1,2,3,\ldots\end{cases}\tag{9.33}$$

From the definition of the $\mathcal{Z}$ transform,

$$F(z)=\sum_{n=0}^{\infty}f[n]z^{-n}=\sum_{n=0}^{\infty}a^nz^{-n}=\sum_{n=0}^{\infty}(az^{-1})^n\tag{9.34}$$

To evaluate this infinite summation, we form a truncated version of $F(z)$ which contains the first k terms of the series. We denote this truncated version as $F_k(z)$. Then,

TABLE 9.1 Properties and Theorems of the $\mathcal{Z}$ transform

Property / Theorem	Time Domain	$\mathcal{Z}$ transform
Linearity	$af_1[n] + bf_2[n] + \ldots$	$aF_1(z) + bF_2(z) + \ldots$
Shift of $x[n]u_0[n]$	$f[n-m]u_0[n-m]$	$z^{-m}F(z)$
Right Shift	$f[n-m]$	$z^{-m}F(z) + \sum_{n=0}^{m-1} f[n-m]z^{-n}$
Left Shift	$f[n+m]$	$z^{m}F(z) + \sum_{n=-m}^{-1} f[n+m]z^{-n}$
Multiplication by a^n	$a^n f[n]$	$F\left(\frac{z}{a}\right)$
Multiplication by e^{-naT}	$e^{-naT}f[n]$	$F(e^{aT}z)$
Multiplication by n	$nf[n]$	$-z\frac{d}{dz}F(z)$
Multiplication by n^2	$n^2 f[n]$	$z\frac{d}{dz}F(z) + z^2\frac{d^2}{dz^2}F(z)$
Summation in Time	$\sum_{m=0}^{n} f[m]$	$\left(\frac{z}{z-1}\right)F(z)$
Time Convolution	$f_1[n]*f_2[n]$	$F_1(z)\cdot F_2(z)$
Frequency Convolution	$f_1[n]\cdot f_2[n]$	$\frac{1}{j2\pi}\oint xF_1(v)F_2\left(\frac{z}{v}\right)v^{-1}dv$
Initial Value Theorem	$f[0] = \lim_{z\to\infty} F(z)$	
Final Value Theorem	$\lim_{n\to\infty} f[n] = \lim_{z\to 1}(z-1)F(z)$	

$$F_k(z) = \sum_{n=0}^{k-1} a^n z^{-n} = 1 + az^{-1} + a^2z^{-2} + \ldots + a^{k-1}z^{-(k-1)} \tag{9.35}$$

and we observe that as $k \to \infty$, (9.35) becomes the same as (9.34).

To express (9.35) in a closed form, we multiply both sides by az^{-1}. Then,

$$az^{-1}F_k(z) = az^{-1} + a^2z^{-2} + a^3z^{-3} + \ldots + a^kz^{-k} \tag{9.36}$$

Subtracting (9.36) from (9.35), we obtain

$$F_k(z) - az^{-1}F_k(z) = 1 - a^kz^{-k}$$

or

$$F_k(z) = \frac{1 - a^kz^{-k}}{1 - az^{-1}} = \frac{1 - (az^{-1})^k}{1 - az^{-1}} \tag{9.37}$$

for $az^{-1} \neq 1$

To determine $F(z)$ from $F_k(z)$, we examine the behavior of the term $(az^{-1})^k$ in the numerator of (9.37). We write the terms az^{-1} and $(az^{-1})^k$ in polar form, that is, $az^{-1} = |az^{-1}|e^{j\theta}$ and

$$(az^{-1})^k = |az^{-1}|^k e^{jk\theta} \tag{9.38}$$

From (9.38) we observe that, for the values of z for which $|az^{-1}| < 1$, the magnitude of the complex number $(az^{-1})^k \to 0$ as $k \to \infty$ and therefore,

$$F(z) = \lim_{k \to \infty} F_k(z) = \frac{1}{1 - az^{-1}} = \frac{z}{z - a} \tag{9.39}$$

for $|az^{-1}| < 1$

For the values of z for which $|az^{-1}| > 1$, the magnitude of the complex number $(az^{-1})^k$ becomes unbounded as $k \to \infty$, and therefore, $F(z) = \lim_{k \to \infty} F_k(z)$ is unbounded for $|az^{-1}| > 1$.

In summary, the transform

$$F(z) = \sum_{n=0}^{\infty} (az^{-1})^n$$

converges to the complex number $z/(z-a)$ for $|az^{-1}| < 1$, and diverges for $|az^{-1}| > 1$. Also, since

$$|az^{-1}| = \left|\frac{a}{z}\right| = \frac{|a|}{|z|}$$

then, $|az^{-1}| < 1$ implies that $|z| > |a|$, while $|az^{-1}| > 1$ implies $|z| < |a|$ and thus,

$$\mathcal{Z}\{a^n u_0[n]\} = \sum_{n=0}^{\infty} a^n z^{-n} = \begin{cases} \dfrac{z}{z-a} & \text{for } |z| > |a| \\ \text{unbounded} & \text{for } |z| < |a| \end{cases} \tag{9.40}$$

The regions of convergence and divergence for the sequence of (9.40) are shown in Figure 9.2.

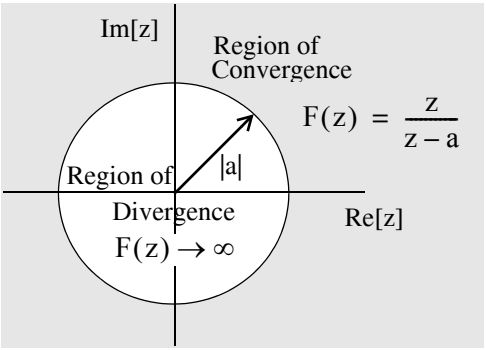


Figure 9.2. Regions of convergence and divergence for the geometric sequence a^n

To determine whether the circumference of the circle, where $|z| = |a|$ |, lies in the region of convergence or divergence, we evaluate the sequence $F_k(z)$ at $z = a$. Then,

$$\begin{aligned} F_k(z) &= \sum_{n=0}^{k-1} a^n z^{-n} = 1 + az^{-1} + a^2 z^{-2} + \ldots + a^{k-1} z^{-(k-1)} \Big|_{z=a} \\ &= 1 + 1 + 1 + \ldots + 1 = k \end{aligned} \tag{9.41}$$

We see that this sequence becomes unbounded as $k \to \infty$, and therefore, the circumference of the circle lies in the region of divergence.

9.3.2 The Transform of the Discrete–Time Unit Step Function

The definition and the waveform of the *discrete–time unit step function* $u_0[n]$ are as shown in Figure 9.3.

$$u_0[n] = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

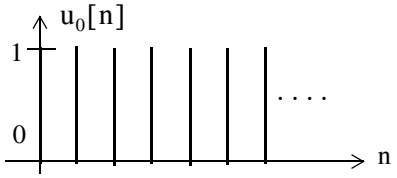


Figure 9.3. The discrete unit step function $u_0[n]$

From the definition of the $\mathcal{Z}$ transform,

$$F(z) = \sum_{n=0}^{\infty} f[n]z^{-n} = \sum_{n=0}^{\infty} [1]z^{-n} \tag{9.42}$$

As in Subsection 9.3.1, to evaluate this infinite summation, we form a truncated version of $F(z)$ which contains the first k terms of the series, and we denote this truncated version as $F_k(z)$. Then,

$$F_k(z) = \sum_{n=0}^{k-1} z^{-n} = 1 + z^{-1} + z^{-2} + \ldots + z^{-(k-1)} \tag{9.43}$$

and we observe that as $k \to \infty$, (9.43) becomes the same as (9.42). To express (9.43) in a closed form, we multiply both sides by z^{-1} and we obtain

$$z^{-1}F_k(z) = z^{-1} + z^{-2} + z^{-3} + \ldots + z^{-k} \tag{9.44}$$

Subtracting (9.44) from (9.43), we obtain

$$F_k(z) - z^{-1}F_k(z) = 1 - z^{-k}$$

or

$$F_k(z) = \frac{1 - z^{-k}}{1 - z^{-1}} = \frac{1 - (z^{-1})^k}{1 - z^{-1}} \tag{9.45}$$

for $z^{-1} \neq 1$

Since $(z^{-1})^k = |z^{-1}|^k e^{jk\theta}$, as $k \to \infty$, $(z^{-1})^k \to 0$. Therefore,

$$F(z) = \lim_{k \to \infty} F_k(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1} \tag{9.46}$$

for $|z| > 1$, and the region of convergence lies outside the unit circle.

Alternate Derivation:

The discrete unit step $u_0[n]$ is a special case of the sequence a^n with $a = 1$, and since $1^n = 1$, by substitution into (9.40) we obtain

$$\mathcal{Z}\{u_0[n]\} = \sum_{n=0}^{\infty} [1]z^{-n} = \begin{cases} \dfrac{z}{z-1} & \text{for} \quad |z| > |1| \\ \text{unbounded} & \text{for} \quad |z| < |1| \end{cases} \tag{9.47}$$

9.3.3 The Transform of the Discrete–Time Exponential Sequence

The *discrete–time exponential sequence* is defined as

$$f[n] = e^{-naT}u_0[n]$$

Then,

$$F(z) = \sum_{n=0}^{\infty} e^{-naT}z^{-n} = 1 + e^{-aT}z^{-1} + e^{-2aT}z^{-2} + e^{-3aT}z^{-3} + \ldots$$

and this is a geometric sequence which can be expressed in closed form as

$$\boxed{\mathcal{Z}[e^{-naT}u_0[n]] = \frac{1}{1-e^{-aT}z^{-1}} = \frac{z}{z-e^{-aT}}} \tag{9.48}$$

for $\left|e^{-aT}z^{-1}\right| < 1$.

9.3.4 The Transform of the Discrete–Time Cosine and Sine Functions

Let

$$f_1[n] = \cos naT$$

and

$$f_2[n] = \sin naT$$

To derive the $\mathcal{Z}$ transform of $f_1[n]$ and $f_2[n]$, we use (9.48) of Subsection 9.3.3, that is,

$$e^{-naT} \Leftrightarrow \frac{z}{z-e^{-aT}}$$

and replacing $-naT$ with $jnaT$ we obtain

$$\mathcal{Z}[e^{jnaT}] = \mathcal{Z}[\cos naT + j\sin naT] = \frac{z}{z-e^{jaT}}$$

$$= \mathcal{Z}[\cos naT] + j\mathcal{Z}[\sin naT] = \frac{z}{z-e^{jaT}} \cdot \frac{z-e^{-jaT}}{z-e^{-jaT}}$$

$$= \mathcal{Z}[\cos naT] + j\mathcal{Z}[\sin naT] = \frac{z^2 - z\cos aT + jz\sin aT}{z^2 - 2z\cos aT + 1}$$

Equating real and imaginary parts, we obtain the transform pairs

$$\cos naT \Leftrightarrow \frac{z^2 - z\cos aT}{z^2 - 2z\cos aT + 1} \tag{9.49}$$

and

$$\sin naT \Leftrightarrow \frac{z\cos aT}{z^2 - 2z\cos aT + 1} \tag{9.50}$$

To define the regions of convergence and divergence, we express the denominator of (9.49) or (9.50) as

$$(z - e^{jaT}) \cdot (z - e^{-jaT}) \tag{9.51}$$

We see that both pairs of (9.49) and (9.50) have two poles, one at $z = e^{jaT}$ and the other at $z = e^{-jaT}$, that is, the poles lie on the unity circle as shown in Figure 9.4.

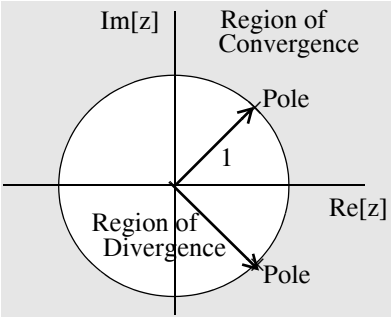


Figure 9.4. Regions of convergence and divergence for $\cos naT$ and $\sin naT$

From Figure 9.4, we see that the poles separate the regions of convergence and divergence. Also, since the circumference of the circle lies on the region of divergence, as we have seen before, the poles lie on the region of divergence. Therefore, for the discrete–time cosine and sine functions we have the pairs

$$\boxed{\cos naT \Leftrightarrow \frac{z^2 - z\cos aT}{z^2 - 2z\cos aT + 1} \quad \text{for } |z| > 1} \tag{9.52}$$

and

$$\boxed{\sin naT \Leftrightarrow \frac{z\sin aT}{z^2 - 2z\cos aT + 1} \quad \text{for } |z| > 1} \tag{9.53}$$

It is shown in complex variables theory that if $F(z)$ is a proper rational function*, all poles lie outside the region of convergence, but the zeros can lie anywhere on the z-plane.

* *This was defined in Chapter 3, page 3–1.*

9.3.5 The Transform of the Discrete–Time Unit Ramp Function

The *discrete–time unit ramp function* is defined as

$$f[n] = nu_0[n]$$

Then,

$$\mathcal{Z}\{nu_0[n]\} = \sum_{n=0}^{\infty} nz^{-n} = 0 + z^{-1} + 2z^{-2} + 3z^{-3} + \ldots \tag{9.54}$$

We can express (9.54) in closed form using the discrete unit step function transform pair

$$\mathcal{Z}\{u_0[n]\} = \sum_{n=0}^{\infty} (1)z^{-n} = \frac{z}{z-1} \quad \text{for} \quad |z| > |1| \tag{9.55}$$

Differentiating both sides of (9.55) with respect to z, we obtain

$$\frac{d}{dz}\left(\sum_{n=0}^{\infty} (1)z^{-n}\right) = \frac{d}{dz}\left(\frac{z}{z-1}\right)$$

or

$$\sum_{n=0}^{\infty} -nz^{-n-1} = \frac{-1}{(z-1)^2}$$

Multiplication by $-z$ yields

$$\sum_{n=0}^{\infty} nz^{-n} = n\sum_{n=0}^{\infty} (1)\, z^{-n} = \frac{z}{(z-1)^2}$$

and thus we have the transform pair

$$\boxed{nu_0[n] \Leftrightarrow \frac{z}{(z-1)^2}} \tag{9.56}$$

We summarize the transform pairs we have derived, and others, given as exercises at the end of this chapter, in Table 9.2.

TABLE 9.2 The $\mathcal{Z}$ transform of common discrete–time functions

f[n]	**F(z)**
$\delta[n]$	1
$\delta[n-m]$	z^{-m}
$a^n u_0[n]$	$\frac{z}{z-a} \quad \lvert z\rvert > a$
$u_0[n]$	$\frac{z}{z-1} \quad \lvert z\rvert > 1$
$(e^{-naT})u_0[n]$	$\frac{z}{z-e^{-aT}} \quad \lvert e^{-aT}z^{-1}\rvert < 1$
$(\cos naT)u_0[n]$	$\frac{z^2 - z\cos aT}{z^2 - 2z\cos aT + 1} \quad \lvert z\rvert > 1$
$(\sin naT)u_0[n]$	$\frac{z\sin aT}{z^2 - 2z\cos aT + 1} \quad \lvert z\rvert > 1$
$(a^n \cos naT)u_0[n]$	$\frac{z^2 - az\cos aT}{z^2 - 2az\cos aT + a^2} \quad \lvert z\rvert > a$
$(a^n \sin naT)u_0[n]$	$\frac{az\sin aT}{z^2 - 2az\cos aT + a^2} \quad \lvert z\rvert > a$
$u_0[n] - u_0[n-m]$	$\frac{z^m - 1}{z^{m-1}(z-1)}$
$nu_0[n]$	$z/(z-1)^2$
$n^2 u_0[n]$	$z(z+1)/(z-1)^3$
$[n+1]u_0[n]$	$z^2/(z-1)^2$
$a^n n u_0[n]$	$(az)/(z-a)^2$
$a^n n^2 u_0[n]$	$az(z+a)/(z-a)^3$
$a^n n[n+1]u_0[n]$	$2az^2/(z-a)^3$

9.4 Computation of the $\mathcal{Z}$ Transform with Contour Integration*

Let $F(s)$ be the Laplace transform of a continuous time function $f(t)$ and $F^*(s)$ the transform of the sampled time function $f(t)$ which we denote as $f^*(t)$. It is shown in complex variables theory that $F^*(s)$ can be derived from $F(s)$ by the use of the *contour integral*

$$F^*(s) = \frac{1}{j2\pi}\oint_C \frac{F(v)}{1-e^{-sT}e^{vT}}\,dv \tag{9.57}$$

where C is a contour enclosing all singularities (poles) of $F(s)$, and v is a dummy variable for s. We can compute the $\mathcal{Z}$ transform of a discrete–time function $f[n]$ using the transformation

$$F(z) = F^*(s)\Big|_{z=e^{sT}} \tag{9.58}$$

By substitution of (9.58) into (9.57), and replacing v with s, we obtain

$$F(z) = \frac{1}{j2\pi}\oint_C \frac{F(s)}{1-z^{-1}e^{sT}}\,ds \tag{9.59}$$

Next, we use *Cauchy's Residue Theorem* to express (9.59) as

$$\boxed{F(z) = \sum_k \text{Res}\frac{F(s)}{1-z^{-1}e^{sT}}\Big|_{s=p_k} = \lim_{s\to p_k}(s-p_k)\frac{F(s)}{1-z^{-1}e^{sT}}} \tag{9.60}$$

Example 9.1

Derive the $\mathcal{Z}$ transform of the discrete unit step function $u_0[n]$ using the residue theorem.

Solution:

We learned in Chapter 2, that

$$\mathcal{L}\,[u_0(t)] = 1/s$$

Then, by residue theorem of (9.60),

* *This section may be skipped without loss of continuity. It is intended for readers who have prior knowledge of complex variables theory. However, the following examples will show that this procedure is not difficult.*

$$F(z) = \lim_{s \to p_k}(s - p_k)\frac{F(s)}{1 - z^{-1}e^{sT}} = \lim_{s \to 0}(s - 0)\frac{1/s}{1 - z^{-1}e^{sT}}$$

$$= \lim_{s \to 0} s\frac{1/s}{1 - z^{-1}e^{sT}} = \lim_{s \to 0}\frac{1}{1 - z^{-1}e^{sT}} = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}$$

for $|z| > 1$, and this is the same as (9.47), Page 9–15.

Example 9.2

Derive the $\mathcal{Z}$ transform of the discrete exponential function $e^{-naT}u_0[n]$ using the residue theorem.

Solution:

From Chapter 2,

$$\mathcal{L}\,[e^{-at}u_0(t)] = \frac{1}{s + a}$$

Then, by residue theorem of (9.60),

$$F(z) = \lim_{s \to p_k}(s - p_k)\frac{F(s)}{1 - z^{-1}e^{sT}} = \lim_{s \to -a}(s + a)\frac{1/(s + a)}{1 - z^{-1}e^{sT}}$$

$$= \lim_{s \to -a}\frac{1}{1 - z^{-1}e^{sT}} = \frac{1}{1 - z^{-1}e^{-aT}} = \frac{z}{z - e^{-aT}}$$

for $|z| > 1$ and this is the same as (9.48), Page 9–16.

Example 9.3

Derive the $\mathcal{Z}$ transform of the discrete unit ramp function $nu_0[n]$ using the residue theorem.

Solution:

From Chapter 2,

$$\mathcal{L}\,[tu_0(t)] = 1/s^2$$

Since $F(s)$ has a second order pole at $s = 0$, we need to apply the residue *theorem applicable to a pole of order n*. This theorem states that

$$\boxed{F(z) = \lim_{s \to p_k}\left(\frac{1}{(n-1)!}\right)(s - p_k)\frac{d^{n-1}}{ds^{n-1}}\left[\frac{F(s)}{1 - z^{-1}e^{sT}}\right]} \qquad (9.61)$$

Thus, for this example,

$$F(z)= \lim_{s\to 0}\frac{d}{ds}\left[s^2\frac{1/s^2}{1-z^{-1}e^{sT}}\right] = \lim_{s\to 0}\frac{d}{ds}\left[\frac{1}{1-z^{-1}e^{sT}}\right] = \frac{z}{(z-1)^2}$$

for $|z|>1$, and this is the same as (9.56), Page 9–18.

9.5 Transformation Between s and z Domains

It is shown in complex variables textbooks that every function of a complex variable maps (transforms) a plane xy to another plane uv. In this section, we will investigate the mapping of the plane of the complex variable s, into the plane of the complex variable z.

Let us reconsider expressions (9.6) and (9.1), Pages 9–2 and 9–1 respectively, which are repeated here for convenience.

$$G(s) = \sum_{n=0}^{\infty} f[n]e^{-nsT} \tag{9.62}$$

and

$$F(z) = \sum_{n=0}^{\infty} f[n]z^{-n} \tag{9.63}$$

By comparison of (9.62) with (9.63),

$$G(s) = F(z)\Big|_{z=e^{sT}} \tag{9.64}$$

Thus, the variables s and z are related as

$$\boxed{z = e^{sT}} \tag{9.65}$$

and

$$\boxed{s = \frac{1}{T}\ln z} \tag{9.66}$$

Therefore,

$$\boxed{F(z) = G(s)\Big|_{s=\frac{1}{T}\ln z}} \tag{9.67}$$

Since s, and z are both complex variables, relation (9.67) allows the mapping (transformation) of regions of the s-plane into the z-plane. We find this transformation by recalling that $s = \sigma + j\omega$ and therefore, expressing z in magnitude-phase form and using (9.65), we obtain

$$z = |z|\angle\theta = |z|e^{j\theta} = e^{\sigma T}e^{j\omega T} \tag{9.68}$$

where,

$$|z| = e^{\sigma T} \tag{9.69}$$

and

$$\theta = \omega T \tag{9.70}$$

Since

$$T = 1/f_s$$

the period T defines the sampling frequency f_s. Then, $\omega_s = 2\pi f_s$ or $f_s = \omega_s/2\pi$, and

$$T = (2\pi)/\omega_s$$

Therefore, we express (9.70) as

$$\theta = \omega\frac{2\pi}{\omega_s} = 2\pi\frac{\omega}{\omega_s} \tag{9.71}$$

and by substitution of (9.69) and (9.71) into (9.68), we obtain

$$\boxed{z = e^{\sigma T}e^{j2\pi(\omega/\omega_s)}} \tag{9.72}$$

The quantity $e^{j2\pi(\omega/\omega_s)}$ in (9.72), defines the unity circle; therefore, let us examine the behavior of z when σ is negative, zero, or positive.

Case I $\sigma < 0$: When σ is negative, from (9.69), we see that $|z| < 1$, and thus the left half of the s-plane maps inside the unit circle of the z-plane, and for different negative values of σ, we obtain concentric circles with radius less than unity.

Case II $\sigma > 0$: When σ is positive, from (9.69), we see that $|z| > 1$, and thus the right half of the s-plane maps outside the unit circle of the z-plane, and for different positive values of σ we obtain concentric circles with radius greater than unity.

Case III $\sigma = 0$: When σ is zero, from (9.72), we see that $z = e^{j2\pi(\omega/\omega_s)}$ and all values of ω lie on the circumference of the unit circle. For illustration purposes, we have mapped several fractional values of the sampling radian frequency ω_s, and these are shown in Table 9.3.

From Table 9.3, we see that the portion of the $j\omega$ axis for the interval $0 \le \omega \le \omega_s$ in the s–plane, maps on the circumference of the unit circle in the z–plane as shown in Figure 9.5. Thus, in digital signal processing the unit circle represents frequencies from zero to the sampling frequency, and the frequency response is the discrete–time transfer function evaluated on the unit circle.

TABLE 9.3 Mapping of multiples of sampling frequency

ω	|z|	θ
0	1	0
$\omega_s/8$	1	$\pi/4$
$\omega_s/4$	1	$\pi/2$
$3\omega_s/8$	1	$3\pi/4$
$\omega_s/2$	1	π
$5\omega_s/8$	1	$5\pi/4$
$3\omega_s/4$	1	$3\pi/2$
$7\omega_s/8$	1	$7\pi/4$
ω_s	1	2π

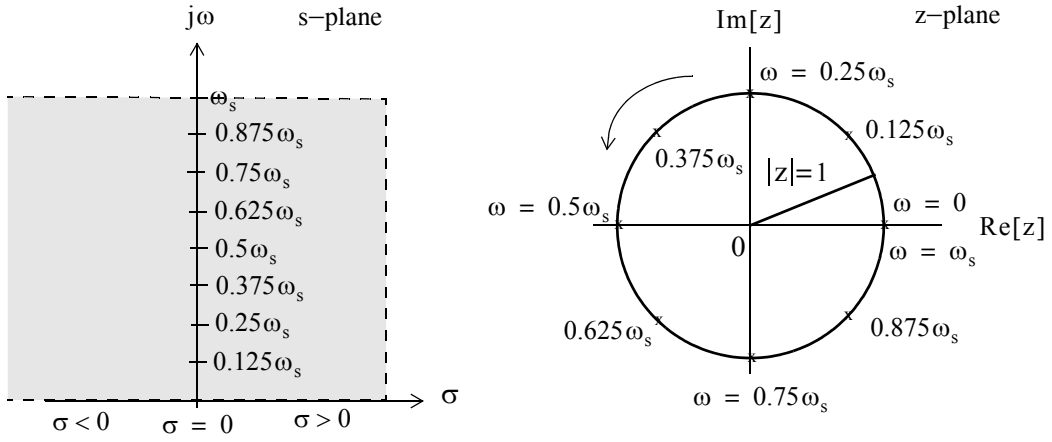


Figure 9.5. Mapping of the s–plane to z–plane

The mapping from the z–plane to the s–plane is a multi–valued transformation since, as we have seen, $s = (1/T)\ln z$, and it is shown in complex variables textbooks that

$$\ln z = \ln z + j2n\pi \tag{9.73}$$

9.6 The Inverse $\mathcal{Z}$ Transform

The Inverse $\mathcal{Z}$ transform enables us to extract $f[n]$ from $F(z)$. It can be found by any of the following three methods:

a. Partial Fraction Expansion

b. The Inversion Integral

c. Long Division of polynomials

These methods are described in Subsections 9.6.1 through 9.6.3 below.

9.6.1 Partial Fraction Expansion

This method is very similar to the partial fraction expansion method that we used in finding the Inverse Laplace transform, that is, we expand $F(z)$ into a summation of terms whose inverse is known. These terms have the form

$$k,\ \frac{r_1 z}{z-p_1},\ \frac{r_2 z}{(z-p_1)^2},\ \frac{r_3 z}{z-p_2},\ \ldots \tag{9.74}$$

where k is a constant, and r_i and p_i represent the residues and poles respectively; these can be real or complex.

Before we expand $F(z)$ into partial fractions, we must express it as a proper rational function. This is done by expanding $F(z)/z$ instead of $F(z)$, that is, we expand it as

$$\frac{F(z)}{z} = \frac{k}{z} + \frac{r_1}{z-p_1} + \frac{r_2}{z-p_2} + \ldots \tag{9.75}$$

and after the residues are found from

$$r_k = \lim_{z \to p_k} (z-p_k)\frac{F(z)}{z} = (z-p_k)\frac{F(z)}{z}\bigg|_{z=p_k} \tag{9.76}$$

we rewrite (9.75) as

$$F(z) = k + \frac{r_1 z}{z-p_1} + \frac{r_2 z}{z-p_2} + \ldots \tag{9.77}$$

Example 9.4

Use the partial fraction expansion method to compute the Inverse $\mathcal{Z}$ transform of

$$F(z) = \frac{1}{(1-0.5z^{-1})(1-0.75z^{-1})(1-z^{-1})} \tag{9.78}$$

Solution:

We multiply both numerator and denominator by z^3 to eliminate the negative powers of z. Then,

$$F(z) = \frac{z^3}{(z-0.5)(z-0.75)(z-1)}$$

Next, we form $F(z)/z$, and we expand in partial fractions as

$$\frac{F(z)}{z} = \frac{z^2}{(z-0.5)(z-0.75)(z-1)} = \frac{r_1}{(z-0.5)} + \frac{r_2}{(z-0.75)} + \frac{r_3}{(z-1)}$$

The residues are

$$r_1 = \left.\frac{z^2}{(z-0.75)(z-1)}\right|_{z=0.5} = \frac{(0.5)^2}{(0.5-0.75)(0.5-1)} = 2$$

$$r_2 = \left.\frac{z^2}{(z-0.5)(z-1)}\right|_{z=0.75} = \frac{(0.75)^2}{(0.75-0.5)(0.75-1)} = -9$$

$$r_3 = \left.\frac{z^2}{(z-0.5)(z-0.75)}\right|_{z=1} = \frac{1^2}{(1-0.5)(1-0.25)} = 8$$

Then,

$$\frac{F(z)}{z} = \frac{z^2}{(z-0.5)(z-0.75)(z-1)} = \frac{2}{(z-0.5)} + \frac{-9}{(z-0.75)} + \frac{8}{(z-1)}$$

and multiplication of both sides by z yields

$$F(z) = \frac{z^3}{(z-0.5)(z-0.75)(z-1)} = \frac{2z}{(z-0.5)} + \frac{-9z}{(z-0.75)} + \frac{8z}{(z-1)} \tag{9.79}$$

To find the Inverse $\mathcal{Z}$ transform of (9.79), we recall that

$$a^n \Leftrightarrow \frac{z}{z-a}$$

for $|z| > a$. Therefore, the discrete–time sequence is

$$f[n] = 2(0.5)^n - 9(0.75)^n + 8 \tag{9.80}$$

Check with MATLAB:

```
Dz=(z–0.5)*(z–0.75)*(z–1)        % The denominator of F(z)
collect(Dz);                     % Multiply the three factors of D(z) to obtain a polynomial
```

```
ans =
  z^3-9/4*z^2+13/8*z-3/8
```

```
num=[0  1  0  0];                % The coefficients of the numerator
den=[1  –9/4   13/8   –3/8];     % The coefficients of the denominator
fprintf(' \n');
[num,den]=residue(num,den);      % Verify the residues in (9.79)
fprintf('r1 = %4.2f \t', num(1)); fprintf('p1 = %4.2f \t', den(1));...
fprintf('r2 = %4.2f \t', num(2)); fprintf('p2 = %4.2f \t', den(2));...
fprintf('r3 = %4.2f \t', num(3)); fprintf('p3 = %4.2f \t', den(3))
```

```
 r1 = 8.00 p1 = 1.00    r2 = -9.00 p2 = 0.75    r3 = 2.00 p3 = 0.50
```

```
syms n z
fn=2*(0.5)^n–9*(0.75)^n+8;       % This is the answer in (9.80)
Fz=ztrans(fn,n,z); simple(Fz)    % Verify answer by first taking Z transform of f[n]
```

```
ans =
 8*z^3/(2*z-1)/(4*z-3)/(z-1)
```

```
iztrans(Fz)                      % Now, verify that Inverse of F(z) gives back f[n]
```

```
ans =
 2*(1/2)^n-9*(3/4)^n+8
```

We can use Microsoft Excel to obtain and plot the values of f[n] . The spreadsheet of Figure 9.6 shows the first 25 values of n but only part of the spreadsheet is shown.

n	f[n]
0.000	1.0000
1.000	2.2500
2.000	3.438
3.000	4.453
4.000	5.277
5.000	5.927
6.000	6.429
7.000	6.814
8.000	7.107
9.000	7.328
10.000	7.495
11.000	7.621
12.000	7.715
13.000	7.786
14.000	7.84
15.000	7.88

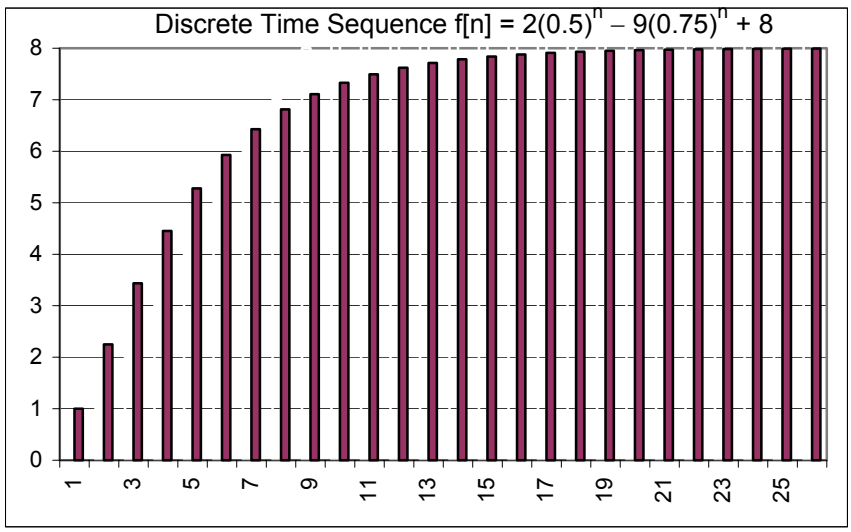


Figure 9.6. The discrete–time sequence $f[n] = 2(0.5)^n - 9(0.75)^n + 8$ for Example 9.4

Example 9.5

Use the partial fraction expansion method to compute the Inverse $\mathcal{Z}$ transform of

$$F(z) = \frac{12z}{(z+1)(z-1)^2} \tag{9.81}$$

Solution:

Division of both sides by z and partial expansion yields

$$\frac{F(z)}{z} = \frac{12}{(z+1)(z-1)^2} = \frac{r_1}{(z+1)} + \frac{r_2}{(z-1)^2} + \frac{r_3}{(z-1)}$$

The residues are

$$r_1 = \left.\frac{12}{(z-1)^2}\right|_{z=-1} = \frac{12}{(-1-1)^2} = 3$$

$$r_2 = \left.\frac{12}{(z+1)}\right|_{z=1} = \frac{12}{(1+1)} = 6$$

$$r_3 = \left.\frac{d}{dz}\left(\frac{12}{z+1}\right)\right|_{z=1} = \frac{-12}{(z+1)^2} = -3$$

Then,

$$\frac{F(z)}{z} = \frac{12}{(z+1)(z-1)^2} = \frac{3}{(z+1)} + \frac{6}{(z-1)^2} + \frac{-3}{(z-1)}$$

or

$$F(z) = \frac{12z}{(z+1)(z-1)^2} = \frac{3z}{(z-(-1))} + \frac{6z}{(z-1)^2} + \frac{-3z}{(z-1)}$$

Now, we recall that

$$u_0[n] \Leftrightarrow \frac{z}{z-1}$$

and

$$nu_0[n] \Leftrightarrow \frac{z}{(z-1)^2}$$

for $|z| > 1$.

Therefore, the discrete–time sequence is

$$f[n] = 3(-1)^n + 6n - 3 \tag{9.82}$$

Check with MATLAB:

```
syms n z; fn=3*(–1)^n+6*n–3; Fz=ztrans(fn); simple(Fz)
```

```
ans =
```

```
12*z/(z+1)/(z-1)^2
```

We can also use the MATLAB **dimpulse** function to compute and display f[n] for any range of values of n. The following script will display the first 20 values of f[n] in (9.82).

```
% First, we must express the denominator of F(z) as a polynomial
denpol=collect((z+1)*((z-1)^2))

denpol =
 z^3-z^2-z+1

num=[12  0];                % The coefficients of the numerator of F(z) in (9.81)
den=[1 -1 -1  1];           % The coefficients of the denominator in polynomial form
fn=dimpulse(num,den,20)     % Compute the first 20 values of f[n]

fn =
      0
      0
     12
     12
     24
     24
     36
     36
     48
     48
     60
     60
     72
     72
     84
     84
     96
     96
    108
    108
```

The MATLAB function **dimpulse(num,den)** plots the impulse response of the polynomial transfer function G(z) = num(z)/den(z) where num(z) and den(z) contain the polynomial coefficients in descending powers of z. Thus, the MATLAB script

```
num=[0  0  12  0]; den=[1 -1 -1  1]; dimpulse(num,den)
```

displays the plot of Figure 9.7.

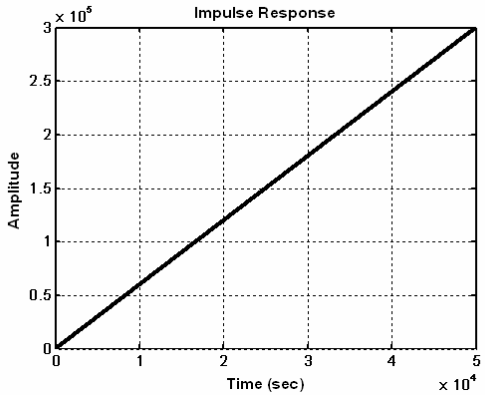


Figure 9.7. The impulse response for Example 9.5

Example 9.6

Use the partial fraction expansion method to compute the Inverse $\mathcal{Z}$ transform of

$$F(z) = \frac{z+1}{(z-1)(z^2+2z+2)} \tag{9.83}$$

Solution:

Dividing both sides by z and performing partial fraction expansion, we obtain

$$\frac{F(z)}{z} = \frac{z+1}{z(z-1)(z^2+2z+2)} = \frac{r_1}{z} + \frac{r_2}{z-1} + \frac{r_3}{(z+1-j)} + \frac{r_4}{(z+1+j)} \tag{9.84}$$

The residues are

$$r_1 = \left.\frac{z+1}{(z-1)(z^2+2z+2)}\right|_{z=0} = \frac{1}{-2} = -0.5$$

$$r_2 = \left.\frac{z+1}{(z)(z^2+2z+2)}\right|_{z=1} = \frac{2}{5} = 0.4$$

$$r_3 = \left.\frac{z+1}{(z)(z-1)(z+1+j)}\right|_{z=-1+j} = \frac{j}{(-1+j)(-2+j)(j2)} = 0.05+j0.15$$

$$r_4 = r^*_3 = 0.05-j0.15$$

Then,

$$\frac{F(z)}{z} = \frac{z+1}{z(z-1)(z^2+2z+2)} = \frac{-0.5}{z} + \frac{0.4}{z-1} + \frac{0.05+j0.15}{(z+1-j)} + \frac{0.05-j0.15}{(z+1+j)}$$

or

$$\begin{aligned} F(z) &= -0.5 + \frac{0.4z}{z-1} + \frac{(0.05+j0.15)z}{(z+1-j)} + \frac{(0.05-j0.15)z}{(z+1+j)} \\ &= -0.5 + \frac{0.4z}{z-1} + \frac{(0.05+j0.15)z}{z-(-1+j)} + \frac{(0.05-j0.15)z}{z-(-1-j)} \\ &= -0.5 + \frac{0.4z}{z-1} + \frac{(0.05+j0.15)z}{z-\sqrt{2}e^{j135°}} + \frac{(0.05-j0.15)z}{z-\sqrt{2}e^{-j135°}} \end{aligned}$$

Recalling that

$$\delta[n] \Leftrightarrow 1$$

and

$$a^n u_0[n] \Leftrightarrow \frac{z}{z-a}$$

for $|z| > a$, we find that the discrete–time sequence is

$$\begin{aligned} f[n] &= -0.5\delta[n] + 0.4(1)^n + (0.05+j0.15)(\sqrt{2}e^{j135°})^n \\ &\quad + (0.05-j0.15)(\sqrt{2}e^{-j135°})^n \\ &= -0.5\delta[n] + 0.4 + 0.05(\sqrt{2}^n e^{jn135°}) + 0.05(\sqrt{2}^n e^{-jn135°}) \\ &\quad + j0.15(\sqrt{2}^n e^{jn135°}) - j0.15(\sqrt{2}^n e^{-jn135°}) \end{aligned}$$

or

$$f[n] = -0.5\delta[n] + 0.4 + \frac{\sqrt{2}^n}{10}\cos n135° - \frac{3\sqrt{2}^n}{10}\sin n135° \tag{9.85}$$

We will use the MATLAB **dimpulse** function to display the first 8 values of $f[n]$ in (9.85). We recall that his function requires that $F(z)$ is expressed as a ratio of polynomials in descending order.

```
syms n z
collect((z–1)*(z^2+2*z+2))   % First, expand denominator of given F(z)
```

```
ans =
  z^3+z^2-2
```

The following script displays the first 10 values of f[n] and plots the impulse response shown in Figure 9.8.

```
num=[0  0  1  1]; den=[1  1  0  –2]; fn=dimpulse(num,den,11), dimpulse(num,den,16)
```

```
fn =
      0
      0
      1
```

```
     0
     0
     2
    -2
     2
    -6
    10
```

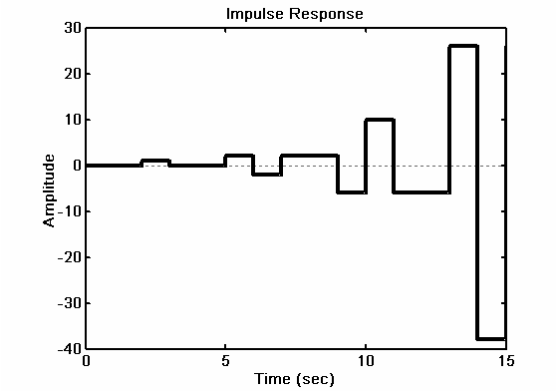


Figure 9.8. The impulse response for Example 9.6

9.6.2 The Inversion Integral

The *inversion integral* * states that

$$f[n] = \frac{1}{j2\pi}\oint_C F(z)z^{n-1}dz \tag{9.86}$$

where C is a closed curve that encloses all poles of the integrant, and by Cauchy's residue theorem, this integral can be expressed as

$$f[n] = \sum_k \text{Res}[F(z)z^{n-1}]\Big|_{z=p_k} \tag{9.87}$$

where p_k represents a pole of $[F(z)z^{n-1}]$ and $\text{Res}[F(z)z^{n-1}]$ represents a residue at $z = p_k$.

* *This section may be skipped without loss of continuity. It is intended for readers who have prior knowledge of complex variables theory.*

Example 9.7

Use the inversion integral method to find the Inverse $\mathcal{Z}$ transform of

$$F(z) = \frac{1 + 2z^{-1} + z^{-3}}{(1 - z^{-1})(1 - 0.75z^{-1})} \tag{9.88}$$

Solution:

Multiplication of the numerator and denominator by z^3 yields

$$F(z) = \frac{z^3 + 2z^2 + 1}{z(z-1)(z-0.75)} \tag{9.89}$$

and by application of (9.87),

$$f[n] = \sum_k \text{Res}\left[\frac{(z^3 + 2z^2 + 1)z^{n-1}}{z(z-1)(z-0.75)}\right]\Bigg|_{z = p_k} = \sum_k \text{Res}\left[\frac{(z^3 + 2z^2 + 1)z^{n-2}}{(z-1)(z-0.75)}\right]\Bigg|_{z = p_k} \tag{9.90}$$

We are interested in the values $f[0], f[1], f[2], \ldots$, that is, values of $n = 0, 1, 2, \ldots$.

For $n = 0$, (9.90) becomes

$$\begin{aligned} f[0] &= \sum_k \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)(z-0.75)}\right]\Bigg|_{z = p_k} \\ &= \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)(z-0.75)}\right]\Bigg|_{z = 0} + \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)(z-0.75)}\right]\Bigg|_{z = 1} \\ &\quad + \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)(z-0.75)}\right]\Bigg|_{z = 0.75} \end{aligned} \tag{9.91}$$

The first term on the right side of (9.91) has a pole of order 2 at $z = 0$; therefore, we must evaluate the first derivative of

$$\frac{(z^3 + 2z^2 + 1)}{(z-1)(z-0.75)}$$

at $z = 0$. Thus, for $n = 0$, (9.91) reduces to

$$f[0] = \frac{d}{dz}\left[\frac{(z^3 + 2z^2 + 1)}{(z-1)(z-0.75)}\right]\Bigg|_{z = 0} + \left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-0.75)}\right]\Bigg|_{z = 1} + \left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)}\right]\Bigg|_{z = 0.75}$$

or

$$f[0] = \frac{28}{9} + 16 - \frac{163}{9} = 1 \tag{9.92}$$

For $n = 1$, (9.90) becomes

$$\begin{aligned} f[1] &= \sum_k \text{Res}\left[\frac{(z^3+2z^2+1)}{z(z-1)(z-0.75)}\right]\Bigg|_{z=p_k} \\ &= \text{Res}\left[\frac{(z^3+2z^2+1)}{z(z-1)(z-0.75)}\right]\Bigg|_{z=0} + \text{Res}\left[\frac{(z^3+2z^2+1)}{z(z-1)(z-0.75)}\right]\Bigg|_{z=1} \\ &\quad + \text{Res}\left[\frac{(z^3+2z^2+1)}{z(z-1)(z-0.75)}\right]\Bigg|_{z=0.75} \\ &= \left[\frac{(z^3+2z^2+1)}{(z-1)(z-0.75)}\right]\Bigg|_{z=0} + \left[\frac{(z^3+2z^2+1)}{z(z-0.75)}\right]\Bigg|_{z=1} + \left[\frac{(z^3+2z^2+1)}{z(z-1)}\right]\Bigg|_{z=0.75} \\ &= \frac{4}{3} + 16 - \frac{163}{12} = \frac{15}{4} \end{aligned} \tag{9.93}$$

For $n \geq 2$ there are no poles at $z = 0$, that is, the only poles are at $z = 1$ and $z = 0.75$. Therefore,

$$\begin{aligned} f[n] &= \sum_k \text{Res}\left[\frac{(z^3+2z^2+1)z^{n-2}}{(z-1)(z-0.75)}\right]\Bigg|_{z=p_k} \\ &= \text{Res}\left[\frac{(z^3+2z^2+1)z^{n-2}}{(z-1)(z-0.75)}\right]\Bigg|_{z=1} + \text{Res}\left[\frac{(z^3+2z^2+1)z^{n-2}}{(z-1)(z-0.75)}\right]\Bigg|_{z=0.75} \\ &= \left[\frac{(z^3+2z^2+1)z^{n-2}}{(z-0.75)}\right]\Bigg|_{z=1} + \left[\frac{(z^3+2z^2+1)z^{n-2}}{(z-1)}\right]\Bigg|_{z=0.75} \end{aligned} \tag{9.94}$$

for $n \geq 2$.

From (9.94), we observe that for all values of $n \geq 2$, the exponential factor z^{n-2} is always unity for $z = 1$, but varies for values $z \neq 1$. Then,

$$\begin{aligned} f[n] &= \left[\frac{(z^3+2z^2+1)}{(z-0.75)}\right]\Bigg|_{z=1} + \left[\frac{(z^3+2z^2+1)z^{n-2}}{(z-1)}\right]\Bigg|_{z=0.75} \\ &= \frac{4}{0.25} + \frac{[0.75^3+2(0.75)^2+1](0.75)^{n-2}}{-0.25} \end{aligned} \tag{9.95}$$

or

$$f[n] = 16 + \frac{(163/64)(0.75)^n}{(-0.25)(0.75)^2} = 16 - \frac{163}{9}(0.75)^n$$

for $n \geq 2$.

We can express $f[n]$ for all $n \geq 0$ as

$$f[n] = \frac{28}{9}\delta[n] + \frac{4}{3}\delta[n-1] + 16 - \frac{163}{9}(0.75)^n \tag{9.96}$$

where the coefficients of $\delta[n]$ and $\delta[n-1]$ are the residues that were found in (9.92) and (9.93) for $n = 0$ and $n = 1$ respectively at $z = 0$. The coefficient $28/9$ is multiplied by $\delta[n]$ to emphasize that this value exists only for $n = 0$ and coefficient $4/3$ is multiplied by $\delta[n-1]$ to emphasize that this value exists only for $n = 1$.

Check with MATLAB:

```
syms z n; Fz=(z^3+2*z^2+1)/(z*(z–1)*(z–0.75)); iztrans(Fz)

ans =
  4/3*charfcn[1](n)+28/9*charfcn[0](n)+16–163/9*(3/4)^n
```

We evaluate and plot $f[n]$ for the first 20 values. This is shown on the spreadsheet of Figure 9.9.

n	f[n]
0	1.000
1	3.750
2	5.813
3	8.359
4	10.270
5	11.702
6	12.777
7	13.582
8	14.187
9	14.640
10	14.980
11	15.235
12	15.426
13	15.570

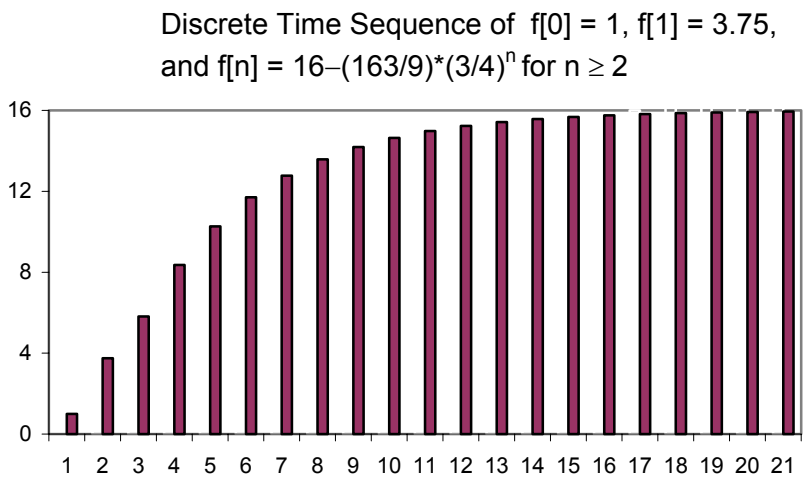


Figure 9.9. The discrete–time sequence for Example 9.7

Example 9.8

Use the inversion integral method to find the Inverse $\mathcal{Z}$ transform of

$$F(z) = \frac{1}{(1-z^{-1})(1-0.75z^{-1})} \tag{9.97}$$

Solution:

Multiplication of the numerator and denominator by z^2 yields

$$F(z) = \frac{z^2}{(z-1)(z-0.75)} \tag{9.98}$$

This function has no poles at $z = 0$. The poles are at $z = 1$ and $z = 0.75$. Then by (9.87),

$$\begin{aligned} f[n] &= \sum_k \text{Res}\left[\frac{z^2 z^{n-1}}{(z-1)(z-0.75)}\right]\Bigg|_{z=p_k} = \sum_k \text{Res}\left[\frac{z^{n+1}}{(z-1)(z-0.75)}\right]\Bigg|_{z=p_k} \\ &= \text{Res}\left[\frac{z^{n+1}}{(z-1)(z-0.75)}\right]\Bigg|_{z=1} + \text{Res}\left[\frac{z^{n+1}}{(z-1)(z-0.75)}\right]\Bigg|_{z=0.75} \\ &= \left[\frac{z^{n+1}}{(z-0.75)}\right]\Bigg|_{z=1} + \left[\frac{z^{n+1}}{(z-1)}\right]\Bigg|_{z=0.75} = \frac{1^{n+1}}{0.25} + \frac{(0.75)^{n+1}}{(-0.25)} \\ &= 4 - \frac{(0.75)^n}{(0.25)(0.75)} = 4 - \frac{16}{3}(0.75)^n \end{aligned} \tag{9.99}$$

9.6.3 Long Division of Polynomials

To apply this method, $F(z)$ must be a rational function, and the numerator and denominator must be polynomials arranged in descending powers of z.

Example 9.9

Use the long division method to determine $f[n]$ for $n = 0, 1,$ and 2, given that

$$F(z) = \frac{1 + z^{-1} + 2z^{-2} + 3z^{-3}}{(1-0.25z^{-1})(1-0.5z^{-1})(1-0.75z^{-1})} \tag{9.100}$$

Solution:

First, we multiply numerator and denominator by z^3, expand the denominator to a polynomial, and arrange the numerator and denominator polynomials in descending powers of z. Then,

$$F(z) = \frac{z^3 + z^2 + 2z + 3}{(z - 0.25)(z - 0.5)(z - 0.75)}$$

Next, we use the MATLAB **collect** function to expand the denominator to a polynomial.

```
syms z; den=collect((z–0.25)*(z–0.5)*(z–0.75))
```

```
den =
  z^3-3/2*z^2+11/16*z-3/32
```

Thus,

$$F(z) = \frac{z^3 + z^2 + 2z + 3}{z^3 - (3/2)z^2 + (11/16)z - 3/32} \tag{9.101}$$

Now, we perform long division as shown in Figure 9.10.

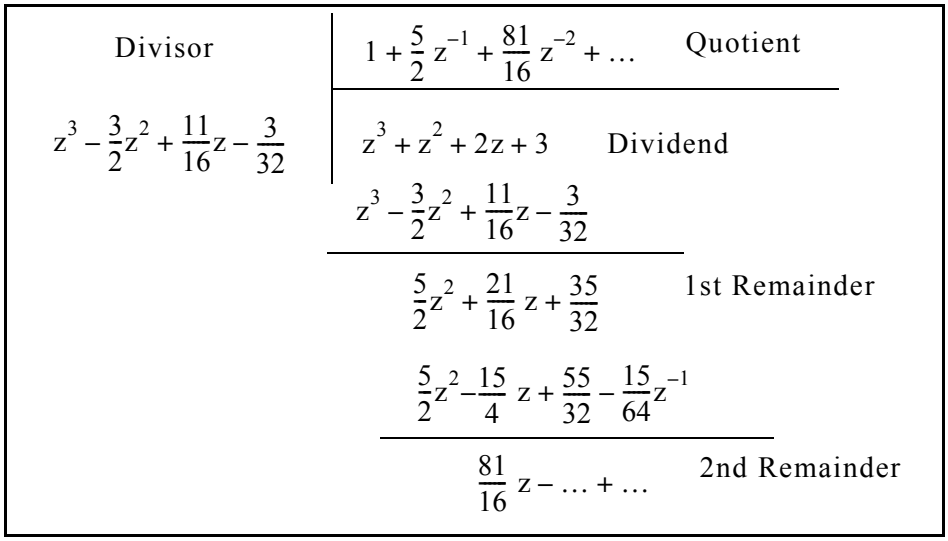


Figure 9.10. Long division for the polynomials of Example 9.9

We find that the quotient $Q(z)$ is

$$Q(z) = 1 + \frac{5}{2}z^{-1} + \frac{81}{16}z^{-2} + \ldots \tag{9.102}$$

By definition of the $\mathcal{Z}$ transform,

$$F(z) = \sum_{n=0}^{\infty} f[n]z^{-n} = f[0] + f[1]z^{-1} + f[2]z^{-2} + \ldots \tag{9.103}$$

Equating like terms in (9.102) and (9.103), we obtain

$$f[0] = 1, \quad f[1] = 5/2 \quad \text{and} \quad f[2] = 81/16 \tag{9.104}$$

We will use the MATLAB **dimpulse** function to verify the answers, and to obtain the sequence of the first 15 values of f[n] .

```
num=[1  1  2  3]; den=[1  –3/2  11/16  –3/32]; fn=dimpulse(num,den,15),...
dimpulse(num,den,16)
```

```
fn =
    1.0000
    2.5000
    5.0625
    8.9688
   10.2070
    9.6191
    8.2522
    6.7220
    5.3115
    4.1195
    3.1577
    2.4024
    1.8189
    1.3727
    1.0338
```

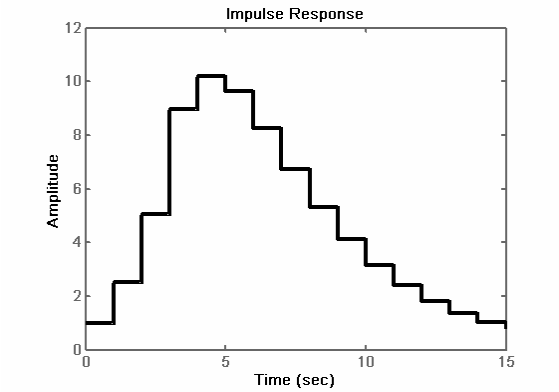


Figure 9.11. Impulse response for Example 9.9

Table 9.4 lists the advantages and disadvantages of the three methods of evaluating the Inverse $\mathcal{Z}$ transform.

9.7 The Transfer Function of Discrete–Time Systems

The discrete–time system of Figure 9.12, can be described by the *linear difference equation*

TABLE 9.4 Methods of Evaluation of the Inverse $\mathcal{Z}$ transform

Method	***Advantages***	***Disadvantages***
Partial Fraction Expansion	• Most familiar • Can use the MATLAB **residue** function	• Requires that $F(z)$ is a proper rational function
Inversion Integral	• Can be used whether $F(z)$ is a rational function or not	• Requires familiarity with the Residues theorem
Long Division	• Practical when only a small sequence of numbers is desired • Useful when Inverse $\mathcal{Z}$ has no closed form solution • Can use the MATLAB **dimpulse** function for large sequence of numbers	• Requires that $F(z)$ is a proper rational function • Division may be endless

Figure 9.12. Block diagram for discrete–time system

$$\begin{aligned} &y[n] + b_1 y[n-1] + b_2 y[n-2] + \ldots + b_k y[n-k] \\ &= a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + \ldots + a_k x[n-k] \end{aligned} \tag{9.105}$$

where a_i and b_i are constant coefficients. In a compact form, relation (9.105) is expressed as

$$y[n] = \sum_{i=0}^{k} a_i x[n-i] - \sum_{i=0}^{k} b_i y[n-i] \tag{9.106}$$

Assuming that all initial conditions are zero, taking the $\mathcal{Z}$ transform of both sides of (9.106), and using the $\mathcal{Z}$ transform pair

$$[f[n-m]] \Leftrightarrow z^{-m} F(z)$$

we obtain

$$\begin{aligned} &Y(z) + b_1 z^{-1} Y(z) + b_2 z^{-2} Y(z) + \ldots + b_k z^{-k} Y(z) \\ &= a_0 X(z) + a_1 z^{-1} X(z) + a_2 z^{-2} X(z) + \ldots + a_k z^{-k} X(z) \end{aligned} \tag{9.107}$$

$$\begin{aligned}(1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_k z^{-k}) Y(z) \\ = (a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_k z^{-k}) X(z)\end{aligned} \tag{9.108}$$

$$Y(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_k z^{-k}}{1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_k z^{-k}} X(z) \tag{9.109}$$

We define the *discrete–time system transfer function* $H(z)$ as

$$H(z) = \frac{N(z)}{D(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_k z^{-k}}{1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_k z^{-k}} \tag{9.110}$$

and by substitution of (9.110) into (9.109), we obtain

$$\boxed{Y(z) = H(z)X(z)} \tag{9.111}$$

The *discrete impulse response* $h[n]$ is the response to the input $x[n] = \delta[n]$, and since

$$\mathcal{Z}\{\delta[n]\} = \sum_{n=0}^{\infty} \delta[n] z^{-n} = 1$$

we can find the discrete–time impulse response $h[n]$ by taking the Inverse $\mathcal{Z}$ transform of the discrete transfer function $H(z)$, that is,

$$\boxed{h[n] = \mathcal{Z}^{-1}\{H(z)\}} \tag{9.112}$$

Example 9.10

The difference equation describing the input–output relationship of a discrete–time system with zero initial conditions, is

$$y[n] - 0.5y[n-1] + 0.125y[n-2] = x[n] + x[n-1] \tag{9.113}$$

Compute:

a. The transfer function $H(z)$

b. The discrete–time impulse response $h[n]$

c. The response when the input is the discrete unit step $u_0[n]$

Solution:

a. Taking the $\mathcal{Z}$ transform of both sides of (9.113), we obtain

$$Y(z)-0.5z^{-1}Y(z)+0.125z^{-2}Y(z) = X(z)+z^{-1}X(z)$$

and thus

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1+z^{-1}}{1-0.5z^{-1}+0.125z^{-2}} = \frac{z^2+z}{z^2-0.5z+0.125} \tag{9.114}$$

b. To obtain the discrete–time impulse response $h[n]$, we need to compute the Inverse $\mathcal{Z}$ transform of (9.114). We first divide both sides by z and we obtain:

$$\frac{H(z)}{z} = \frac{z+1}{z^2-0.5z+0.125} \tag{9.115}$$

Using the MATLAB **residue** function, we obtain the residues and the poles of (9.115) as follows:

```
num=[0  1  1]; den=[1  –0.5  0.125]; [num,den]=residue(num,den);fprintf(' \n');...
disp('r1 = '); disp(num(1)); disp('p1 = '); disp(den(1));...
disp('r2 = '); disp(num(2)); disp('p2 = '); disp(den(2))
```

```
r1 =
   0.5000 - 2.5000i
p1 =
   0.2500 + 0.2500i
r2 =
   0.5000 + 2.5000i
p2 =
   0.2500 - 0.2500i
```

and thus,

$$\frac{H(z)}{z} = \frac{0.5-j2.5}{z-0.25-j0.25}+\frac{0.5+j2.5}{z-0.25+j0.25}$$

or

$$H(z) = \frac{(0.5-j2.5)z}{z-(0.25+j0.25)}+\frac{(0.5+j2.5)z}{z-(0.25-j0.25)} = \frac{(0.5-j2.5)z}{z-0.25\sqrt{2}e^{j45°}}+\frac{(0.5+j2.5)z}{z-0.25\sqrt{2}e^{-j45°}} \tag{9.116}$$

Recalling that

$$a^n u_0[n] \Leftrightarrow \frac{z}{z-a}$$

for $|z|>a$, the discrete impulse response sequence $h[n]$ is

$$\begin{aligned} h[n] &= (0.5-j2.5)(0.25\sqrt{2}e^{\,j45^\circ})^n + (0.5+j2.5)(0.25\sqrt{2}e^{-j45^\circ})^n \\ &= 0.5[(0.25\sqrt{2})^n e^{\,jn45^\circ}] + 0.5[(0.25\sqrt{2})^n e^{-jn45^\circ}] \\ &\quad -j2.5[(0.25\sqrt{2})^n e^{\,jn45^\circ}] + j2.5[(0.25\sqrt{2})^n e^{-jn45^\circ}] \\ &= 0.5[(0.25\sqrt{2})^n (e^{\,jn45^\circ} + e^{-jn45^\circ})] - j2.5(0.25\sqrt{2})^n (e^{\,jn45^\circ} - e^{-jn45^\circ}) \end{aligned}$$

or

$$h[n] = \left(\frac{\sqrt{2}}{4}\right)^n (\cos n45^\circ + 5\sin n45^\circ) \tag{9.117}$$

c. From $Y(z) = H(z)X(z)$, the transform $u_0[n] \Leftrightarrow \dfrac{z}{z-1}$, and using the result of part (a) we obtain:

$$Y(z) = \frac{z^2+z}{z^2-0.5z+0.125} \cdot \frac{z}{z-1} = \frac{z(z^2+z)}{(z^2-0.5z+0.125)(z-1)}$$

or

$$\frac{Y(z)}{z} = \frac{(z^2+z)}{(z^2-0.5z+0.125)(z-1)} \tag{9.118}$$

We will use the MATLAB **residue** function to compute the residues and poles of expression (9.117). First, we must express the denominator as a polynomial.

```
syms z; denom=(z^2–0.5*z+0.125)*(z–1); collect(denom)
```

```
ans =
 z^3-3/2*z^2+5/8*z-1/8
```

Then,

$$\frac{Y(z)}{z} = \frac{z^2+z}{z^3-(3/2)z^2+(5/8)z-1/8} \tag{9.119}$$

Now, we compute the residues and poles.

```
num=[0  1  1  0]; den=[1  –3/2  5/8  –1/8]; [num,den]=residue(num,den); fprintf(' \n');...
disp('r1 = '); disp(num(1)); disp('p1 = '); disp(den(1));...
disp('r2 = '); disp(num(2)); disp('p2 = '); disp(den(2));...
disp('r3 = '); disp(num(3)); disp('p3 = '); disp(den(3))
```

```
r1 =
    3.2000
p1 =
    1.0000
r2 =
```

```
   -1.1000 + 0.3000i
p2 =
    0.2500 + 0.2500i
r3 =
   -1.1000 - 0.3000i
p3 =
    0.2500 - 0.2500i
```

With these values, we express (9.119) as

$$\frac{Y(z)}{z} = \frac{z^2+z}{z^3-(3/2)z^2+(5/8)z-1/8} = \frac{3.2}{z-1}+\frac{-1.1+j0.3}{z-0.25-j0.25}+\frac{-1.1-j0.3}{z-0.25+j0.25} \tag{9.120}$$

or

$$\begin{aligned} Y(z) &= \frac{3.2z}{z-1}+\frac{(-1.1+j0.3)z}{z-0.25-j0.25}+\frac{(-1.1-j0.3)z}{z-0.25+j0.25} \\ &= \frac{3.2z}{z-1}+\frac{(-1.1+j0.3)z}{z-0.25\sqrt{2}e^{j45°}}+\frac{(-1.1-j0.3)z}{z-0.25\sqrt{2}e^{-j45°}} \end{aligned} \tag{9.121}$$

Recalling that

$$a^n u_0[n] \Leftrightarrow \frac{z}{z-a}$$

for $|z| > a|$, we find that the discrete output response sequence is

$$\begin{aligned} y[n] &= 3.2+(-1.1+j0.3)(0.25\sqrt{2}e^{j45°})^n-(1.1+j0.3)(0.25\sqrt{2}e^{-j45°})^n \\ &= 3.2-1.1[(0.25\sqrt{2})^n(e^{jn45°}+e^{-jn45°})]+j0.3[(0.25\sqrt{2})^n(e^{jn45°}-e^{-jn45°})] \end{aligned}$$

or

$$\begin{aligned} y[n] &= 3.2-2.2\left(\frac{\sqrt{2}}{4}\right)^n \cos n45°-0.6\left(\frac{\sqrt{2}}{4}\right)^n \sin n45° \\ &= 3.2-\left(\frac{\sqrt{2}}{4}\right)^n (2.2\cos n45°+0.6\sin n45°) \end{aligned} \tag{9.122}$$

The plots for the discrete–time sequences h[n] and y[n] are shown in Figure 9.13.

The plot for y[n] can be readily obtained with the Simulink model shown in Figure 9.14 where in the Function Block Parameters dialog box for the **Discrete Transfer Fcn** block the Numerator and Denominator coefficients were specified as [1 1 0] and [1 −0.5 0.125] respectively in accordance with the transfer function of relation (9.114), Page 9–41, where we found that

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1+z^{-1}}{1-0.5z^{-1}+0.125z^{-2}} = \frac{z^2+z}{z^2-0.5z+0.125}$$

n	h[n]	y[n]
0.000	1.000	1.000
1.000	1.500	2.500
2.000	0.625	3.125
3.000	0.125	3.250
4.000	-0.02	3.234
5.000	-0.02	3.211
6.000	-0.010	3.201
7.000	-0	3.199
8.000	0.000	3.199
9.000	0.000	3.200
10.000	0.000	3.200
11.000	3E-05	3.200
12.000	-0	3.200
13.000	-0	3.200

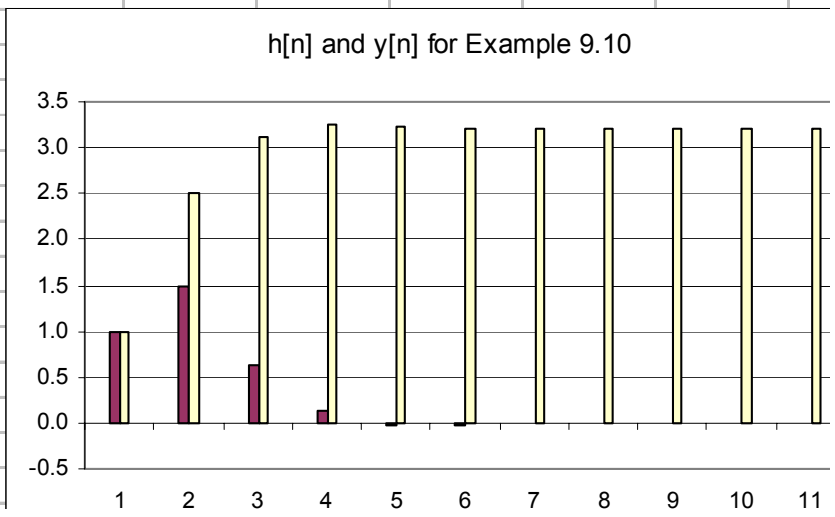


Figure 9.13. Plots of h[n] and y[n] for Example 9.10

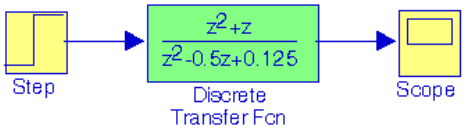


Figure 9.14. Model for Example 9.10

The plot for y[n], displayed on the Scope block of Figure 9.14, is shown in Figure 9.15.

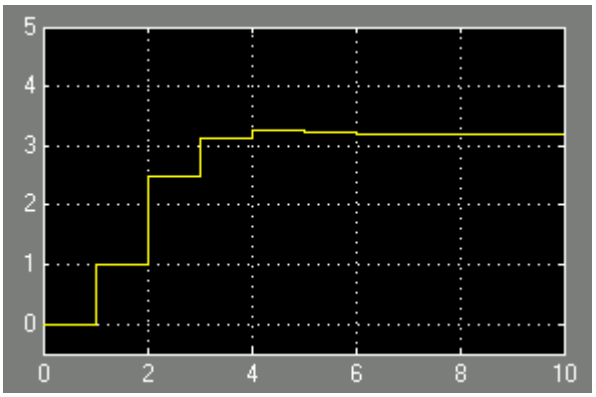


Figure 9.15. Output waveform for the model of Figure 9.14

9.8 State Equations for Discrete–Time Systems

As with continuous time systems, we choose *state variables for discrete–time systems*, either from block diagrams that relate the input–output information, or directly from a difference equation.

Consider the block diagram of Figure 9.16.

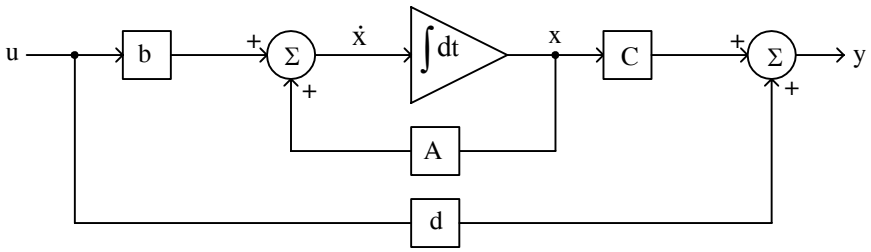


Figure 9.16. Block diagram for a continuous time system

We learned in Chapter 5 that the state equations representing this continuous time system are

$$\begin{aligned} \dot{x} &= Ax + bu \\ y &= Cx + du \end{aligned} \tag{9.123}$$

In a discrete–time block diagram, the integrator is replaced by a delay device. The analogy between an integrator and a unit delay device is shown in Figure 9.17.

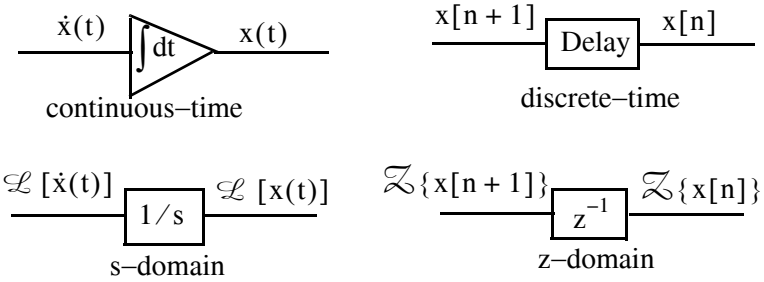


Figure 9.17. Analogy between integration and delay devices

Example 9.11

The input–output relation for a discrete–time system is

$$y[n+3] + 2y[n+2] + 5y[n+1] + y[n] = u[n] \tag{9.124}$$

where $u[n]$ is any input, and $y[n]$ is the output. Write the discrete–time state equations for this system.

Solution:

We choose our state variables as the output and the output advanced by one and by two time steps. Thus, we choose the discrete state variables as

$$x_1[n] = y[n] \qquad x_2[n] = y[n+1] \qquad x_3[n] = y[n+2] \tag{9.125}$$

Then,

$$x_3[n+1] = y[n+3]$$

$$x_2[n+1] = y[n+2] = x_3[n]$$

$$x_1[n+1] = y[n+1] = x_2[n]$$

Thus, the state equations are

$$\begin{aligned} x_1[n+1] &= x_2[n] \\ x_2[n+1] &= x_3[n] \\ x_3[n+1] &= -2x_3[n] - 5x_2[n] - x_1[n] = u[n] \end{aligned}$$

and in matrix form,

$$\begin{bmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -5 & -2 \end{bmatrix} \cdot \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u[n] \tag{9.126}$$

The general form of the solution is

$$x[n] = A^n x[0] + \sum_{i=0}^{n-1} A^{n-1-i} b[i]u[i] \tag{9.127}$$

The discrete–time state equations are written in a more compact form as

$$\begin{aligned} x[n+1] &= Ax[n] + bu[n] \\ y[n] &= Cx[n] + du[n] \end{aligned} \tag{9.128}$$

We can use the MATLAB **c2d** function to convert the continuous–time state–space equation

$$\dot{x}(t) = Ax(t) + bu(t) \tag{9.129}$$

to the discrete–time state space equation

$$x[n+1] = A_{disc}\, x[n] + b_{disc} u[n] \tag{9.130}$$

where the subscript disc stands for discrete, n indicates the present sample, and n + 1 indicates the next sample.

Example 9.12

Use the MATLAB **c2d** function to convert the continuous time state space equation

$$\dot{x}(t) = Ax(t) + bu(t)$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} \text{ and } b = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{9.131}$$

to discrete–time state space equation with sampling period $T_S = 0.1$ s.

Solution:

```
Adisc=[0  1; –3  –4]; bdisc=[0  1]'; [Adisc,bdisc]=c2d(Adisc,bdisc,0.1)
```

```
Adisc =
    0.9868    0.0820
   -0.2460    0.6588
bdisc =
    0.0044
    0.0820
```

and therefore, the equivalent discrete–time state–space equation is

$$\begin{bmatrix} x_1[n+1] \\ x_2[n+1] \end{bmatrix} = \begin{bmatrix} 0.9868 & 0.0820 \\ -0.2460 & 0.6588 \end{bmatrix} \cdot \begin{bmatrix} x_1[n] \\ x_2[n] \end{bmatrix} + \begin{bmatrix} 0.0044 \\ 0.0820 \end{bmatrix} u[n] \tag{9.132}$$

The MATLAB **d2c** function converts the discrete–time state equation

$$x[n+1] = A_{disc}x[n] + b_{disc}u[n]$$

to the continuous time state equation

$$\dot{x}(t) = Ax(t) + bu(t)$$

We can invoke the MATLAB command **help d2c** to obtain a detailed description of this function.

9.9 Summary

- The $\mathcal{Z}$ transform performs the transformation from the domain of discrete–time signals, to another domain which we call $z-\text{domain}$. It is used with discrete–time signals, the same way the Laplace and Fourier transforms are used with continuous–time signals.
- The one-sided $\mathcal{Z}$ transform $F(z)$ of a discrete–time function $f[n]$ defined as

$$F(z) = \sum_{n=0}^{\infty} f[n]z^{-n}$$

and it is denoted as

$$F(z) = \mathcal{Z}\{f[n]\}$$

- The Inverse $\mathcal{Z}$ transform is defined as

$$f[n] = \frac{1}{j2\pi}\oint F(z)z^{k-1}dz$$

and it is denoted as

$$f[n] = \mathcal{Z}^{-1}\{F(z)\}$$

- The linearity property of the $\mathcal{Z}$ transform states that

$$af_1[n] + bf_2[n] + cf_3[n] + \ldots \Leftrightarrow aF_1(z) + bF_2(z) + cF_3(z) + \ldots$$

- The shifting of $f[n]u_0[n]$ where $u_0[n]$ is the discrete unit step function, produces the $\mathcal{Z}$ transform pair

$$f[n-m]u_0[n-m] \Leftrightarrow z^{-m}F(z)$$

- The right shifting of $f[n]$ allows use of non-zero values for $n<0$ and produces the $\mathcal{Z}$ transform pair

$$f[n-m] \Leftrightarrow z^{-m}F(z) + \sum_{n=0}^{m-1} f[n-m]z^{-n}$$

For $m = 1$, this transform pair reduces to

$$f[n-1] \Leftrightarrow z^{-1}F(z) + f[-1]$$

and for $m = 2$, reduces to

$$f[n-2] \Leftrightarrow z^{-2}F(z) + f[-2] + z^{-1}f[-1]$$

- The mth left shifting of f[n] where m is a positive integer, produces the $\mathcal{Z}$ transform pair

$$f[n+m] \Leftrightarrow z^m F(z) + \sum_{n=-m}^{-1} f[n+m]z^{-n}$$

For $m = 1$, the above expression reduces to

$$\mathcal{Z}\{f[n+1]\} = zF(z) - f[0]z$$

and for $m = 2$, reduces to

$$\mathcal{Z}\{f[n+2]\} = z^2F(z) - f[0]z^2 - f[1]z$$

- Multiplication by a^n produces the $\mathcal{Z}$ transform pair

$$a^n f[n] \Leftrightarrow F\left(\frac{z}{a}\right)$$

- Multiplication by e^{-naT} produces the $\mathcal{Z}$ transform pair

$$e^{-naT} f[n] \Leftrightarrow F(e^{aT}z)$$

- Multiplications by n and n^2 produce the $\mathcal{Z}$ transform pairs

$$nf[n] \Leftrightarrow -z\frac{d}{dz}F(z)$$

$$n^2 f[n] \Leftrightarrow z\frac{d}{dz}F(z) + z^2\frac{d^2}{dz^2}F(z)$$

- The summation property of the $\mathcal{Z}$ transform states that

$$\sum_{m=0}^{n} f[m] \Leftrightarrow \left(\frac{z}{z-1}\right)F(z)$$

- Convolution in the discrete–time domain corresponds to multiplication in the z-domain, that is,

$$f_1[n] * f_2[n] \Leftrightarrow F_1(z) \cdot F_2(z)$$

- Multiplication in the discrete–time domain corresponds to convolution in the z-domain, that is,

$$f_1[n] \cdot f_2[n] \Leftrightarrow \frac{1}{j2\pi}\oint xF_1(v)F_2\left(\frac{z}{v}\right)v^{-1}dv$$

- The initial value theorem of the $\mathcal{Z}$ transform states that

$$f[0] = \lim_{z \to \infty} X(z)$$

- The final value theorem of the $\mathcal{Z}$ transform states that

$$\lim_{n \to \infty} f[n] = \lim_{z \to 1} (z-1)F(z)$$

- The $\mathcal{Z}$ transform of the geometric sequence

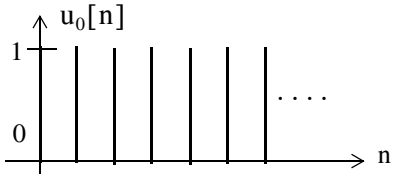
$$f[n] = \begin{cases} 0 & n = -1, -2, -3, \ldots \\ a^n & n = 0, 1, 2, 3, \ldots \end{cases}$$

is

$$\mathcal{Z}[a^n] = \sum_{n=0}^{\infty} a^n z^{-n} = \begin{cases} \dfrac{z}{z-a} & \text{for} \quad |z| > |a| \\ \text{unbounded} & \text{for} \quad |z| < |a| \end{cases}$$

- The $\mathcal{Z}$ transform of the discrete unit step function $u_0[n]$ shown below

$$u_0[n] = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$$



is

$$\mathcal{Z}[u_0[n]] = \sum_{n=0}^{\infty} [1] z^{-n} = \begin{cases} \dfrac{z}{z-1} & \text{for} \quad |z| > |1| \\ \text{unbounded} & \text{for} \quad |z| < |1| \end{cases}$$

- The $\mathcal{Z}$ transform of the discrete exponential sequence

$$f[n] = e^{-naT}$$

is

$$\mathcal{Z}[e^{-naT}] = \frac{1}{1 - e^{-aT} z^{-1}} = \frac{z}{z - e^{-aT}}$$

for $\left| e^{-aT} z^{-1} \right| < 1$

- The $\mathcal{Z}$ transforms of the discrete–time functions $f_1[n] = \cos naT$ and $f_2[n] = \sin naT$ are respectively

$$\cos naT \Leftrightarrow \frac{z^2 - z\cos aT}{z^2 - 2z\cos aT + 1} \quad \text{for } |z| > 1$$

$$\sin naT \Leftrightarrow \frac{z\sin aT}{z^2 - 2z\cos aT + 1} \quad \text{for } |z| > 1$$

- The $\mathcal{Z}$ transform of the discrete unit ramp $f[n] = nu_0[n]$ is

$$nu_0[n] \Leftrightarrow \frac{z}{(z-1)^2}$$

- The $\mathcal{Z}$ transform can also be found by means of the contour integral

$$F^*(s) = \frac{1}{j2\pi}\oint_C \frac{F(v)}{1 - e^{-sT}e^{vT}}\,dv$$

and the residue theorem.

- The variables s and z are related as

$$z = e^{sT}$$

and

$$s = \frac{1}{T}\ln z$$

- The relation

$$F(z) = G(s)\Big|_{s = \frac{1}{T}\ln z}$$

allows the mapping (transformation) of regions of s-plane to z-plane.

- The Inverse $\mathcal{Z}$ transform can be found by partial fraction expansion, the inversion integral, and long division of polynomials.

- The discrete–time system transfer function $H(z)$ is defined as

$$H(z) = \frac{N(z)}{D(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_k z^{-k}}{1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_k z^{-k}}$$

- The input $X(z)$ and output $Y(z)$ are related by the system transfer function $H(z)$ as

$$Y(z) = H(z)X(z)$$

- The discrete–time impulse response $h[n]$ and the discrete transfer function $H(z)$ are related as

$$h[n] = \mathcal{Z}^{-1}\{H(z)\}$$

- The discrete–time state equations are

$$x[n+1] = Ax[n] + bu[n]$$
$$y[n] = Cx[n] + du[n]$$

and the general form of the solution is

$$x[n] = A^n x[0] + \sum_{i=0}^{n-1} A^{n-1-i} b[i]u[i]$$

- The MATLAB **c2d** function converts the continuous time state space equation

$$\dot{x}(t) = Ax(t) + bu(t)$$

to the discrete–time state space equation

$$x[n+1] = A_{disc}\,x[n] + b_{disc}u[n]$$

- The MATLAB **d2c** function converts the discrete–time state equation

$$x[n+1] = A_{disc}x[n] + b_{disc}u[n]$$

to the continuous time state equation

$$\dot{x}(t) = Ax(t) + bu(t)$$

9.10 Exercises

1. Find the $\mathcal{Z}$ transform of the discrete–time pulse $p[n]$ defined as

$$p[n] = \begin{cases} 1 & n = 0, 1, 2, \ldots, m-1 \\ 0 & \text{otherwise} \end{cases}$$

2. Find the $\mathcal{Z}$ transform of $a^n p[n]$ where $p[n]$ is defined as in Exercise 1.

3. Prove the following $\mathcal{Z}$ transform pairs:

a. $\delta[n] \Leftrightarrow 1$ b. $\delta[n-1] \Leftrightarrow z^{-m}$ c. $na^n u_0[n] \Leftrightarrow \dfrac{az}{(z-a)^2}$

d. $n^2 a^n u_0[n] \Leftrightarrow \dfrac{az(z+a)}{(z-a)^3}$ e. $[n+1]u_0[n] \Leftrightarrow \dfrac{z^2}{(z-1)^2}$

4. Use the partial fraction expansion to find $f[n] = \mathcal{Z}^{-1}[F(z)]$ given that

$$F(z) = \frac{A}{(1-z^{-1})(1-0.5z^{-1})}$$

5. Use the partial fraction expansion method to compute the Inverse $\mathcal{Z}$ transform of

$$F(z) = \frac{z^2}{(z+1)(z-0.75)^2}$$

6. Use the Inversion Integral to compute the Inverse $\mathcal{Z}$ transform of

$$F(z) = \frac{1+2z^{-1}+z^{-3}}{(1-z^{-1})(1-0.5z^{-1})}$$

7. Use the long division method to compute the first 5 terms of the discrete–time sequence whose $\mathcal{Z}$ transform is

$$F(z) = \frac{z^{-1}+z^{-2}-z^{-3}}{1+z^{-1}+z^{-2}+4z^{-3}}$$

8.

a. Compute the transfer function of the difference equation

$$y[n] - y[n-1] = Tx[n-1]$$

b. Compute the response $y[n]$ when the input is $x[n] = e^{-naT}$

9. Given the difference equation

$$y[n] - y[n-1] = \frac{T}{2}\{x[n] + x[n-1]\}$$

a. Compute the discrete transfer function $H(z)$

b. Compute the response to the input $x[n] = e^{-naT}$

10. A discrete–time system is described by the difference equation

$$y[n] + y[n-1] = x[n]$$

where

$$y[n] = 0 \quad \text{for} \quad n < 0$$

a. Compute the transfer function $H(z)$

b. Compute the impulse response $h[n]$

c. Compute the response when the input is $x[n] = 10 \quad \text{for} \quad n \geq 0$

11. Given the discrete transfer function

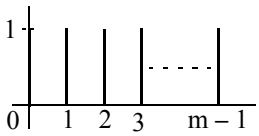
$$H(z) = \frac{z+2}{8z^2 - 2z - 3}$$

write the difference equation that relates the output $y[n]$ to the input $x[n]$.

9.11 Solutions to End–of–Chapter Exercises

1.

$$p[n] = \begin{cases} 1 & n = 0, 1, 2, \ldots, m-1 \\ 0 & \text{otherwise} \end{cases}$$



$$p[n] = u_0[n] - u_0[n-m]$$

$$u_0[n] \Leftrightarrow \frac{z}{z-1}$$

$$u_0[n-m] \Leftrightarrow z^{-m}\frac{z}{z-1}$$

By the linearity property

$$\mathcal{Z}\{p[n]\} = \frac{z}{z-1} - z^{-m}\frac{z}{z-1} = \frac{z(1-z^{-m})}{z-1} = \frac{1-z^{-m}}{1-z^{-1}}$$

2.

$$a^n f[n] \Leftrightarrow F\left(\frac{z}{a}\right)$$

and from Exercise 1,

$$p[n] \Leftrightarrow \frac{1-z^{-m}}{1-z^{-1}}$$

Then,

$$a^n p[n] \Leftrightarrow \frac{1-(z/a)^{-m}}{1-(z/a)^{-1}} = \frac{(a^{-m}-z^{-m})/a^{-m}}{(a^{-1}-z^{-1})/a^{-1}} = \frac{a^{-1}(a^{-m}-z^{-m})}{a^{-m}(a^{-1}-z^{-1})} = \frac{a^m(a^{-m}-z^{-m})}{a(a^{-1}-z^{-1})} = \frac{1-a^m z^{-m}}{1-az^{-1}}$$

or

$$a^n p[n] \Leftrightarrow \frac{z(1-a^m z^{-m})}{z-a}$$

3.

a.

$$\mathcal{Z}\{\delta[n]\} = \sum_{n=0}^{\infty} \delta[n]z^{-n} = \delta[0]z^{-0} = 1$$

b.

$$\mathcal{Z}\{\delta[n-m]\} = \sum_{n=0}^{\infty} \delta[n-m]z^{-n}$$

and since $\delta[n-m]$ is zero for all n except $n = m$, it follows that

$$\mathcal{Z}\{\delta[n-m]\} = \sum_{n=0}^{\infty} \delta[0]z^{-m} = z^{-m}$$

c.

From (9.40), Page 9–14,

$$f[n] = a^n u_0[n] \Leftrightarrow F(z) = \frac{z}{z-a} \quad (1)$$

Differentiating (1) with respect to z and multiplying by $-z$ we obtain

$$-z\frac{d}{dz}F(z) = -z\frac{z-a-z}{(z-a)^2} = \frac{az}{(z-a)^2} \quad (2)$$

Also, from the multiplication by n property

$$nf[n] = n(a^n u_0[n]) \Leftrightarrow -z\frac{d}{dz}F(z) \quad (3)$$

and from (2) and (3)

$$n(a^n u_0[n]) \Leftrightarrow \frac{az}{(z-a)^2} \quad (4)$$

we observe that for $a = 1$ (4) above reduces to

$$nu_0[n] \Leftrightarrow \frac{z}{(z-1)^2}$$

d.

From (9.40), Page 9–14,

$$f[n] = a^n u_0[n] \Leftrightarrow F(z) = \frac{z}{z-a} \quad (1)$$

and taking the second derivative of (1) with respect to z we obtain

$$\frac{d^2}{dz^2}F(z) = \frac{d^2}{dz^2}\left(\frac{z}{z-a}\right) = \frac{d}{dz}\left[\frac{d}{dz}\left(\frac{z}{z-a}\right)\right] = \frac{d}{dz}\left[\frac{-a}{(z-a)^2}\right] = \frac{2a(z-a)}{(z-a)^4} = \frac{2a}{(z-a)^3} \quad (2)$$

Also, from the multiplication by n^2 property

$$n^2 f[n] = n^2(a^n u_0[n]) \Leftrightarrow z\frac{d}{dz}F(z) + z^2\frac{d^2}{dz^2}F(z) \quad (3)$$

From Exercise 9.3(c), relation (2)

$$z\frac{d}{dz}F(z) = \frac{-az}{(z-a)^2} \quad (4)$$

and by substitution of (2) and (4) into (3) we obtain

$$n^2(a^n u_0[n]) \Leftrightarrow \frac{-az}{(z-a)^2} + z^2\frac{2a}{(z-a)^3} = \frac{2az^2 - az(z-a)}{(z-a)^3} = \frac{2az^2 - az^2 + a^2 z}{(z-a)^3} = \frac{az(z+a)}{(z-a)^3}$$

We observe that for $a = 1$ the above reduces to

$$n^2 u_0[n] \Leftrightarrow \frac{z(z+1)}{(z-1)^3}$$

e.

Let $f[n] = u_0[n]$ and we know that

$$u_0[n] \Leftrightarrow \frac{z}{z-1} \quad (1)$$

The term $(n+1)u_0[n]$ represents the sum of the first n values, including $n = 0$, of $u_0[n]$ and thus it can be written as the summation

$$g[n] = (n+1)u_0[n] = \sum_{k=0}^{n} u_0[k]$$

Since summation in the discrete–time domain corresponds to integration in the continuous time domain, it follows that

$$u_1[n] = (n+1)u_0[n]$$

where $u_1[n]$ represents the discrete unit ramp. Now, from the summation in time property,

$$\sum_{k=0}^{n} f[k] \Leftrightarrow \left(\frac{z}{z-1}\right)F(z)$$

and with (1) above

$$G(z) = \left(\frac{z}{z-1}\right)F(z) = \frac{z}{z-1}\cdot\frac{z}{z-1} = \frac{z^2}{(z-1)^2}$$

and thus

$$[n+1]u_0[n] \Leftrightarrow \frac{z^2}{(z-1)^2}$$

4.

First, we multiply the numerator and denominator by z^2 to eliminate the negative exponents of z.

Then,

$$F(z) = \frac{A}{(1-z^{-1})(1-0.5z^{-1})} = \frac{Az^2}{(z-1)(z-0.5)}$$

or

$$\frac{F(z)}{z} = \frac{Az}{(z-1)(z-0.5)} = \frac{r_1}{z-1} + \frac{r_2}{z-0.5}$$

$$r_1 = \left.\frac{Az}{z-0.5}\right|_{z=1} = 2A \qquad r_2 = \left.\frac{Az}{z-1}\right|_{z=0.5} = -A$$

$$\frac{F(z)}{z} = \frac{2A}{z-1} - \frac{A}{z-0.5}$$

$$F(z) = \frac{2Az}{z-1} - \frac{Az}{z-0.5}$$

Since

$$\frac{z}{z-1} \Leftrightarrow 1 \qquad \frac{z}{z-a} \Leftrightarrow a^n$$

it follows that

$$\mathcal{Z}^{-1}[F(z)] = f[n] = 2A - A\left(\frac{1}{2}\right)^n = A\left[2 - \left(\frac{1}{2}\right)^n\right]$$

5.

$$\frac{F(z)}{z} = \frac{r_1}{z+1} + \frac{r_2}{z-0.75} + \frac{r_3}{(z-0.75)^2} = \frac{z}{(z+1)(z-0.75)^2} \quad (1)$$

and clearing of fractions yields

$$r_1(z-0.75)^2 + r_2(z+1)(z-0.75) + r_3(z+1) = z \quad (2)$$

With $z = 0.75$ (2) reduces to $1.75r_3 = 0.75$ from which $r_3 = 3/7$

With $z = -1$ (2) reduces to $(-1.75)^2 r_1 = -1$ from which $r_1 = -16/49$

With $z = 0$ (2) reduces to $(-0.75)^2 r_1 - 0.75r_2 + r_3 = 0$

or $(3/4)^2 \times (-16/49) - (3/4)r_2 + 3/7 = 0$ from which $r_2 = 16/49$

By substitution into (1) and multiplication by z we obtain

$$F(z) = \frac{(-16/49)z}{z-(-1)} + \frac{(16/49)z}{z-0.75} + \frac{(4/7)\cdot(0.75z)}{(z-0.75)^2}$$

Using the transforms $u_0[n] \Leftrightarrow \frac{z}{z-1}$, $a^n u_0[n] = \frac{z}{z-a}$, and $na^n u_0[n] = \frac{az}{(z-a)^2}$ we obtain

$$f[n] = \left(-\frac{16}{49}\right)(-1)^n + \left(\frac{16}{49}\right)(0.75)^n + \frac{4}{7}n(0.75)^n$$

Check with MATLAB:

```
syms z n; Fz=z^2/((z+1)*(z–0.75)^2); iztrans(Fz)
```

```
ans =
   -16/49*(-1)^n+16/49*(3/4)^n+4/7*(3/4)^n*n
```

6.

Multiplication by z^3 yields

$$F(z) = \frac{z^3 + 2z^2 + 1}{z(z-1)(z-0.5)}$$

From (9.87), Page 9–32,

$$f[n] = \sum_k \text{Res}[F(z)z^{n-1}]\Big|_{z=p_k}$$

and for this exercise,

$$f[n] = \sum_k \text{Res}\left[\frac{(z^3 + 2z^2 + 1)z^{n-2}}{(z-1)(z-0.5)}\right]\Big|_{z=p_k}$$

Next, we examine z^{n-2} to find out if there are any values of n for which there is a pole at the origin. We observe that for $n = 0$ there is a second order pole at $z = 0$ because

$$z^{n-2}\Big|_{n=0} = z^{-2} = \frac{1}{z^2}$$

Also, for $n = 1$ there is a simple pole at $z = 0$. But for $n \geq 2$ the only poles are $z = 1$ and $z = 0.5$. Then, following the same procedure as in Example 9.12, for $n = 0$ we obtain:

$$f[0] = \sum_k \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)(z-0.5)}\right]\Big|_{z=p_k}$$

$$= \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)(z-0.5)}\right]\Big|_{z=0} + \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)(z-0.5)}\right]\Big|_{z=1} + \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)(z-0.5)}\right]\Big|_{z=0.5}$$

The first term on the right side of the above expression has a pole of order 2 at $z = 0$; therefore, we must evaluate the first derivative of

$$\frac{(z^3 + 2z^2 + 1)}{(z-1)(z-0.5)}$$

at $z = 0$. Thus, for $n = 0$, it reduces to

$$f[0] = \frac{d}{dz}\left[\frac{(z^3 + 2z^2 + 1)}{(z-1)(z-0.5)}\right]\Bigg|_{z=0} + \left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-0.5)}\right]\Bigg|_{z=1} + \left[\frac{(z^3 + 2z^2 + 1)}{z^2(z-1)}\right]\Bigg|_{z=0.5}$$

$$= 6 + 8 - 13 = 1$$

For $n = 1$, it reduces to

$$f[1] = \sum_k \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z(z-1)(z-0.5)}\right]\Bigg|_{z=p_k}$$

$$= \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z(z-1)(z-0.5)}\right]\Bigg|_{z=0} + \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z(z-1)(z-0.5)}\right]\Bigg|_{z=1} + \text{Res}\left[\frac{(z^3 + 2z^2 + 1)}{z(z-1)(z-0.75)}\right]\Bigg|_{z=0.5}$$

or

$$f[1] = \left[\frac{(z^3 + 2z^2 + 1)}{(z-1)(z-0.75)}\right]\Bigg|_{z=0} + \left[\frac{(z^3 + 2z^2 + 1)}{z(z-0.75)}\right]\Bigg|_{z=1} + \left[\frac{(z^3 + 2z^2 + 1)}{z(z-1)}\right]\Bigg|_{z=0.5}$$

$$= 2 + 8 - 13 \cdot (0.5) = 3.5$$

For $n \geq 2$ there are no poles at $z = 0$, that is, the only poles are at $z = 1$ and $z = 0.5$. Therefore,

$$f[n] = \sum_k \text{Res}\left[\frac{(z^3 + 2z^2 + 1)z^{n-2}}{(z-1)(z-0.5)}\right]\Bigg|_{z=p_k}$$

$$= \text{Res}\left[\frac{(z^3 + 2z^2 + 1)z^{n-2}}{(z-1)(z-0.5)}\right]\Bigg|_{z=1} + \text{Res}\left[\frac{(z^3 + 2z^2 + 1)z^{n-2}}{(z-1)(z-0.5)}\right]\Bigg|_{z=0.5}$$

$$= \left[\frac{(z^3 + 2z^2 + 1)z^{n-2}}{(z-0.5)}\right]\Bigg|_{z=1} + \left[\frac{(z^3 + 2z^2 + 1)z^{n-2}}{(z-1)}\right]\Bigg|_{z=0.5}$$

for $n \geq 2$.

We can express $f[n]$ for all $n \geq 0$ as

$$f[n] = 6\delta[n] + 2\delta[n-1] + 8 - 13(0.5)^n$$

where the coefficients of $\delta[n]$ and $\delta[n-1]$ are the residues that were found for $n = 0$ and $n = 1$ at $z = 0$. The coefficient 6 is multiplied by $\delta[n]$ to emphasize that this value exists only for $n = 0$ and coefficient 2 is multiplied by $\delta[n]$ to emphasize that this value exists only for $n = 1$.

Check with MATLAB:

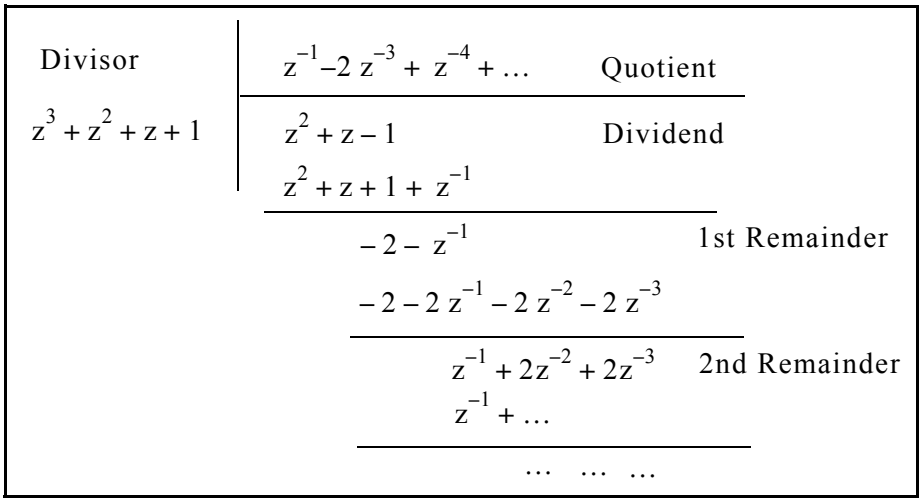
```
syms z n; Fz=(z^3+2*z^2+1)/(z*(z–1)*(z–0.5)); iztrans(Fz)

ans =
 2*charfcn[1](n)+6*charfcn[0](n)+8–13*(1/2)^n
```

7.

Multiplication of each term by z^3 yields

$$F(z) = \frac{z^{-1} + z^{-2} - z^{-3}}{1 + z^{-1} + z^{-2} + 4z^{-3}} = \frac{z^2 + z - 1}{z^3 + z^2 + z + 1}$$

The long division of the numerator by the denominator is shown below.



Therefore,

$$F(z) = z^{-1} - 2z^{-3} + z^{-4} + \ldots \quad (1)$$

Also,

$$F(z) = \sum_0^\infty f[n]z^{-n} = f[0] + f[1]z^{-1} + f[2]z^{-2} + f[3]z^{-3} + f[4]z^{-4} + \ldots \quad (2)$$

Equating like terms on the right sides of (1) and (2) we obtain

$$f[0] = 0 \qquad f[1] = 1 \qquad f[2] = 0 \qquad f[3] = -2 \qquad f[4] = 1$$

8.

a.

$$y[n] - y[n-1] = Tx[n-1]$$

Taking the $\mathcal{Z}$ transform of both sides we obtain

$$Y(z) - z^{-1}Y(z) = Tz^{-1}X(z)$$

and thus

$$H(z) = \frac{Y(z)}{X(z)} = \frac{Tz^{-1}}{1 - z^{-1}} = \frac{T}{z-1}$$

b.

$$x[n] = e^{-naT} \Leftrightarrow X(z) = \frac{z}{z - e^{-aT}}$$

Then,

$$Y(z) = H(z)X(z) = \frac{T}{z-1} \cdot \frac{z}{z - e^{-aT}} = \frac{Tz}{(z-1) \cdot (z - e^{-aT})}$$

or

$$\frac{Y(z)}{z} = \frac{T}{(z-1) \cdot (z - e^{-aT})} = \frac{r_1}{z-1} + \frac{r_2}{z - e^{-aT}} \quad (1)$$

$$r_1 = \left.\frac{T}{z - e^{-aT}}\right|_{z=1} = \frac{T}{1 - e^{-aT}} \qquad r_2 = \left.\frac{T}{z-1}\right|_{z = e^{-aT}} = \frac{-T}{1 - e^{-aT}}$$

By substitution into (1) and multiplication by z we obtain

$$Y(z) = \frac{Tz/(1 - e^{-aT})}{(z-1)} - \frac{Tz/(1 - e^{-aT})}{(z - e^{-aT})}$$

Recalling that $\frac{z}{z-1} \Leftrightarrow u_0[n]$ and $\frac{z}{z-a} \Leftrightarrow a^n u_0[n]$ we obtain

$$y[n] = \frac{T}{1 - e^{-aT}} - \frac{Te^{-naT}}{1 - e^{-aT}} = \frac{T}{1 - e^{-aT}}(1 - e^{-naT})u_0[n]$$

9.

a.

$$y[n] - y[n-1] = \frac{T}{2}\{x[n] + x[n-1]\}$$

Taking the $\mathcal{Z}$ transform of both sides we obtain

$$Y(z) - z^{-1}Y(z) = \frac{T}{2}[X(z) + z^{-1}X(z)]$$

$$(1 - z^{-1})Y(z) = \frac{T}{2}(1 + z^{-1})X(z)$$

and thus

$$H(z) = \frac{Y(z)}{X(z)} = \frac{T}{2} \cdot \frac{1 + z^{-1}}{1 - z^{-1}} = \frac{T}{2} \cdot \frac{z + 1}{z - 1}$$

b.

$$x[n] = e^{-naT} \Leftrightarrow X(z) = \frac{z}{z - e^{-aT}}$$

Then,

$$Y(z) = H(z)X(z) = \frac{T}{2} \cdot \frac{z + 1}{z - 1} \cdot \frac{z}{z - e^{-aT}} = \frac{Tz(z + 1)}{2(z - 1) \cdot (z - e^{-aT})}$$

or

$$\frac{Y(z)}{z} = \frac{T(z + 1)}{2(z - 1) \cdot (z - e^{-aT})} = \frac{r_1}{z - 1} + \frac{r_2}{z - e^{-aT}} \quad (1)$$

$$r_1 = \frac{T}{2} \cdot \frac{z + 1}{z - e^{-aT}}\bigg|_{z = 1} = \frac{T}{2} \cdot \frac{2}{1 - e^{-aT}} = \frac{T}{1 - e^{-aT}}$$

$$r_2 = \frac{T}{2} \cdot \frac{z + 1}{z - 1}\bigg|_{z = e^{-aT}} = \frac{T}{2} \cdot \frac{e^{-aT} + 1}{e^{-aT} - 1} = \frac{T}{1 - e^{-aT}}$$

By substitution into (1) and multiplication by z we obtain

$$Y(z) = \frac{Tz/(1 - e^{-aT})}{(z - 1)} + \frac{[(Tz/2) \cdot (e^{-aT} + 1)]/(1 - e^{-aT})}{(z - e^{-aT})}$$

Recalling that $\frac{z}{z - 1} \Leftrightarrow u_0[n]$ and $\frac{z}{z - a} \Leftrightarrow a^n u_0[n]$ we obtain

$$y[n] = \frac{T}{1 - e^{-aT}} + \frac{T}{2} \cdot \frac{e^{-aT} + 1}{e^{-aT} - 1} e^{-naT} = \frac{T}{1 - e^{-aT}} - \frac{T}{2} \coth\left(\frac{aT}{2}\right) e^{-naT}$$

10.

a.

$$y[n] + y[n - 1] = x[n]$$

$$y[n] = 0 \quad \text{for} \quad n < 0$$

Taking the $\mathcal{Z}$ transform of both sides we obtain

$$(1+z^{-1})Y(z) = X(z)$$

and thus

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1+z^{-1}} = \frac{z}{z+1} = \frac{z}{z-(-1)}$$

b.

$$h[n] = \mathcal{Z}^{-1}\{H(z)\} = \mathcal{Z}^{-1}\left\{\frac{z}{z-(-1)}\right\} = (-1)^n$$

c.

$$x[n] = 10 \quad \text{for} \quad n \geq 0$$

$$X(z) = 10 \cdot \frac{z}{z-1}$$

$$Y(z) = H(z)X(z) = \frac{z}{z+1} \cdot \frac{10z}{z-1} = \frac{10z^2}{(z+1)(z-1)}$$

$$\frac{Y(z)}{z} = \frac{10z}{(z+1)(z-1)} = \frac{r_1}{z+1} + \frac{r_2}{z-1} = \frac{5}{z+1} + \frac{5}{z-1}$$

$$Y(z) = \frac{5z}{z-(-1)} + \frac{5z}{z-1} \Leftrightarrow f[n] = 5(-1)^n + 5$$

11.

$$H(z) = \frac{z+2}{8z^2-2z-3}$$

Multiplication of each term by $1/8z^2$ yields

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1/8 \cdot (z^{-1}+2z^{-2})}{1-(1/4)z^{-1}-(3/8)z^{-2}}$$

$$\left[1-\frac{1}{4}z^{-1}-\frac{3}{8}z^{-2}\right] \cdot Y(z) = \frac{1}{8} \cdot (z^{-1}+2z^{-2}) \cdot X(z)$$

and taking the Inverse $\mathcal{Z}$ transform, we obtain

$$y[n]-\frac{1}{4}y[n-1]-\frac{3}{8}y[n-2] = \frac{1}{8}x[n-1]+\frac{1}{4}x[n-2]$$

Chapter 10

The DFT and the FFT Algorithm

This chapter begins with the actual computation of frequency spectra for discrete time systems. For brevity, we will use the acronyms DFT for the Discrete Fourier Transform and FFT for Fast Fourier Transform algorithm respectively. The definition, theorems, and properties are also discussed, and several examples are presented to illustrate their uses.

10.1 The Discrete Fourier Transform (DFT)

In the Fourier series topic, Chapter 7, we learned that a periodic and continuous time function, results in a non periodic and discrete frequency function. Next, in the Fourier transform topic, Chapter 8, we saw that a non–periodic and continuous time function, produces a non–periodic and continuous frequency function. In Chapter 9 we learned that the $\mathcal{Z}$ and Inverse $\mathcal{Z}$ transforms produce a periodic and continuous frequency function, since these transforms are evaluated on the unit circle. This is because the frequency spectrum of a discrete time sequence $f[n]$ is obtained from its $\mathcal{Z}$ transform by the substitution of $z = e^{sT} = e^{j\omega T}$ as we saw from the mapping of the s–plane to the z–plane in Chapter 9, Page 9–23. It is continuous because there is an infinite number of points in the interval 0 to 2π, although, in practice, we compute only a finite number of equally spaced points.

In this chapter we will see that a periodic and discrete time function results in a periodic and discrete frequency function. For convenience, we summarize these facts in Table 10.1.

TABLE 10.1 Characteristics of Fourier and $\mathcal{Z}$ transforms

Topic	***Time Function***	***Frequency Function***
Fourier Series	Continuous, Periodic	Discrete, Non–Periodic
Fourier Transform	Continuous, Non–Periodic	Continuous, Non–Periodic
$\mathcal{Z}$ transform	Discrete, Non–Periodic	Continuous, Periodic
Discrete Fourier Transform	Discrete, Periodic	Discrete, Periodic

In our subsequent discussion we will denote a discrete time signal as $x[n]$, and its discrete frequency transform as $X[m]$.

Let us consider again the definition of the $\mathcal{Z}$ transform, that is,

$$F(z) = \sum_{n=0}^{\infty} f[n]z^{-n} \tag{10.1}$$

Its value on the unit circle of the z-plane, is

$$F(e^{j\omega T}) = \sum_{n=0}^{\infty} f[n]e^{-nj\omega T} \tag{10.2}$$

This represents an infinite summation; it must be truncated before it can be computed. Let this truncated version be represented by

$$\boxed{X[m] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{mn}{N}}} \tag{10.3}$$

where N represents the number of points that are equally spaced in the interval 0 to 2π on the unit circle of the z-plane, and

$$\omega = \left(\frac{2\pi}{NT}\right)m$$

for $m = 0, 1, 2, \ldots, N-1$. We refer to relation (10.3) as the $N-$point DFT of $X[m]$.

The Inverse DFT is defined as

$$\boxed{x[n] = \frac{1}{N}\sum_{m=0}^{N-1} X[m]e^{j2\pi\frac{mn}{N}}} \tag{10.4}$$

for $n = 0, 1, 2, \ldots, N-1$.

In general, the discrete frequency transform $X[m]$ is complex, and thus we can express it as

$$X[m] = Re\{X[m]\} + Im\{X[m]\} \tag{10.5}$$

for $m = 0, 1, 2, \ldots, N-1$.

Since

$$e^{-j2\pi\frac{mn}{N}} = \cos\frac{2\pi mn}{N} - j\sin\frac{2\pi mn}{N} \tag{10.6}$$

we can express (10.3) as

$$X[m] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{mn}{N}} = \sum_{n=0}^{N-1} x[n]\cos\frac{2\pi mn}{N} - j\sum_{n=0}^{N-1} x[n]\sin\frac{2\pi mn}{N} \tag{10.7}$$

For $n = 0$, (10.7) reduces to $X[m] = x[0]$. Then, the real part $\mathrm{Re}\{X[m]\}$ can be computed from

$$\mathrm{Re}\{X[m]\} = x[0] + \sum_{n=1}^{N-1} x[n]\cos\frac{2\pi mn}{N} \quad \text{for} \quad m = 0, 1, 2, \ldots, N-1 \tag{10.8}$$

and the imaginary part $\mathrm{Im}\{X[m]\}$ from

$$\mathrm{Im}\{X[m]\} = -\sum_{n=1}^{N-1} x[n]\sin\frac{2\pi mn}{N} \quad \text{for} \quad m = 0, 1, 2, \ldots, N-1 \tag{10.9}$$

We observe that the summation in (10.8) and (10.9) is from $n = 1$ to $n = N-1$ since $x[0]$ appears in (10.8).

Example 10.1

A discrete time signal is defined by the sequence

$$x[0] = 1, \quad x[1] = 2, \quad x[2] = 2, \quad \text{and} \quad x[3] = 1$$

and $x[n] = 0$ for all other n. Compute the frequency components $X[m]$.

Solution:

Since we are given four discrete values of $x[n]$, we will use a 4–point DFT, that is, for this example, $N = 4$. Using (10.8) with $n = 0, 1, 2,$ and 3, we obtain

$$\begin{aligned}\mathrm{Re}\{X[m]\} &= x[0] + \sum_{n=1}^{3} x[n]\cos\frac{2\pi mn}{N} \\ &= 1 + 2\cos\frac{2\pi m(1)}{4} + 2\cos\frac{2\pi m(2)}{4} + \cos\frac{2\pi m(3)}{4} \\ &= 1 + 2\cos\frac{m\pi}{2} + 2\cos m\pi + \cos\frac{3m\pi}{2}\end{aligned} \tag{10.10}$$

Next, for $m = 0, 1, 2,$ and 3, we obtain:

$$\begin{aligned} m &= 0 \quad \mathrm{Re}\{X[0]\} = 1 + 2\cdot(1) + 2\cdot(1) + 1\cdot(1) = 6 \\ m &= 1 \quad \mathrm{Re}\{X[1]\} = 1 + 2\cdot(0) + 2\cdot(-1) + 1\cdot(0) = -1 \\ m &= 2 \quad \mathrm{Re}\{X[2]\} = 1 + 2\cdot(-1) + 2\cdot(1) + 1\cdot(-1) = 0 \\ m &= 3 \quad \mathrm{Re}\{X[3]\} = 1 + 2\cdot(0) + 2\cdot(-1) + 1\cdot(0) = -1 \end{aligned} \tag{10.11}$$

Now, we compute the imaginary components using (10.9). For this example,

$$\mathrm{Im}\{X[m]\} = -\sum_{n=1}^{3} x[n]\sin\frac{2\pi mn}{N} = -2\sin\frac{m\pi}{2} - 2\sin m\pi - \sin\frac{3m\pi}{2}$$

and for $m = 0, 1, 2,$ and 3, we obtain:

$$\begin{aligned}
m &= 0 \qquad \mathrm{Im}\{X[0]\} = -2\cdot(0) - 2\cdot(0) - 1\cdot(0) = 0 \\
m &= 1 \qquad \mathrm{Im}\{X[1]\} = -2\cdot(1) - 2\cdot(0) - 1\cdot(-1) = -1 \\
m &= 2 \qquad \mathrm{Im}\{X[2]\} = -2\cdot(0) - 2\cdot(0) - 1\cdot(0) = 0 \\
m &= 3 \qquad \mathrm{Im}\{X[3]\} = -2\cdot(-1) - 2\cdot(0) - 1\cdot(1) = 1
\end{aligned} \tag{10.12}$$

The discrete frequency components $X[m]$ for $m = 0, 1, 2,$ and 3 are found by addition of the real and imaginary components $X_{re}[i]$ of (10.11) and $X_{im}[i]$ of (10.12). Thus,

$$\begin{aligned}
X[0] &= 6 + j0 = 6 \\
X[1] &= -1 - j \\
X[2] &= 0 + j0 = 0 \\
X[3] &= -1 + j
\end{aligned} \tag{10.13}$$

Example 10.2

Use the Inverse DFT, i.e., relation (10.4), and the results of Example 10.1, to compute the values of the discrete time sequence $x[n]$.

Solution:

Since we are given four discrete values of $x[n]$, we will use a 4–point DFT, that is, for this example, $N = 4$. Then, (10.4) for $m = 0, 1, 2,$ and 3, reduces to

$$\begin{aligned}
x[n] &= \frac{1}{4}\sum_{m=0}^{3} X[m]e^{j2\pi\frac{mn}{4}} = \frac{1}{4}\sum_{m=0}^{3} X[m]e^{j\pi\frac{mn}{2}} \\
&= \frac{1}{4}\left[X[0] + X[1]e^{j\pi\frac{n}{2}} + X[2]e^{j\pi n} + X[3]e^{j\pi\frac{3n}{2}}\right]
\end{aligned} \tag{10.14}$$

The discrete frequency components $x[n]$ for $n = 0, 1, 2,$ and 3, are:

$$\begin{aligned}
x[0] &= \frac{1}{4}\{X[0] + X[1] + X[2] + X[3]\} \\
&= \frac{1}{4}\{6 + (-1 - j) + 0 + (-1 + j)\} = 1
\end{aligned}$$

$$x[1] = \frac{1}{4}\{X[0] + X[1]\cdot j + X[2]\cdot(-1) + X[3]\cdot(-j)\}$$

$$= \frac{1}{4}\{6 + (-1-j)\cdot j + 0\cdot(-1) + (-1+j)\cdot(-j)\} = 2$$

$$x[2] = \frac{1}{4}\{X[0] + X[1]\cdot(-1) + X[2]\cdot 1 + X[3]\cdot(-1)\}$$

$$= \frac{1}{4}\{6 + (-1-j)\cdot(-1) + 0\cdot 1 + (-1+j)\cdot(-1)\} = 2$$

$$x[3] = \frac{1}{4}\{X[0] + X[1]\cdot(-j) + X[2]\cdot(-1) + X[3]\cdot j\}$$

$$= \frac{1}{4}\{6 + (-1-j)\cdot(-j) + 0\cdot(-1) + (-1+j)\cdot j\} = 1$$

We observe that these are the same values as in Example 10.1.

We will check the answers of Examples 10.1 and 10.2 with MATLAB and Excel.

With MATLAB, we use the **fft(x)** function to compute the DFT, and the **ifft(x)** function to compute the Inverse DFT.

```
xn=[1  2  2  1];          % The discrete time sequence of Example 10.1
Xm=fft(xn)                % Compute the FFT of this discrete time sequence
```

```
Xm =
   6.0000   -1.0000-1.0000i   0   -1.0000+1.0000i
```

```
Xm = [6  −1−j  0  −1+j];     % The discrete frequency components of Example 10.2
xn=ifft(Xm)                  % Compute the Inverse FFT
```

```
xn =
   1.0000  2.0000+0.0000i  2.0000  1.0000-0.0000i
```

To use Excel for the computation of the DFT, the *Analysis ToolPak* must have been installed. If not, it can installed it by clicking *Add–Ins* on the *Tools* drop menu, and following the instructions on the screen.

With Excel's Fourier Analysis Tool, we get the spreadsheet shown in Figure 10.1. The instructions on how to use it, are given on the spreadsheet.

The term $e^{-(j2\pi)/N}$ is a rotating vector, where the range $0 \le \theta \le 2\pi$ is divided into $360/N$ equal segments. Therefore, it is convenient to represent it as W_N, that is, we let

$$W_N = e^{-\frac{j2\pi}{N}} \tag{10.15}$$

	A	B	C	D	E
1	Input data x(n) are same as in Example 10.1				
2	and are entered in cells A11 through A14				
3					
4	From the Tools drop down menu, we select				
5	Data Analysis and from it, Fourier Analysis				
6					
7	The Input Range is A11 through A14 (A11:A14)				
8	and the Output Range is B11 through B14 (B11:B14				
9					
10	**x(n)**	**X(m)**			
11	1	6			
12	2	-1-i			
13	2	0			
14	1	-1+i			
15					
16	To obtain the discrete time sequence, we select				
17	Inverse from the Fourier Analysis menu				
18					
19	Input data are the same as in Example 10.2				
20					
21	The Input Range is A25 through A28 (A25:A28)				
22	and the Output Range is B25 through B28 (B25:B28				
23					
24	**X(m)**	**x(n)**			
25	6	1			
26	-1-j	2			
27	0	2			
28	-1+j	1			

Figure 10.1. Using Excel to find the DFT and Inverse DFT

and consequently

$$W_N^{-1} = e^{\frac{j2\pi}{N}} \tag{10.16}$$

Henceforth, the DFT pair will be denoted as

$$\boxed{X[m] = \sum_{n=0}^{N-1} x(n) W_N^{mn}} \tag{10.17}$$

and

$$\boxed{x[n] = \frac{1}{N} \sum_{n=0}^{N-1} X(m) W_N^{-mn}} \tag{10.18}$$

Also, the correspondence between x[n] and X[m] will be denoted as

$$x[n] \Leftrightarrow X[m] \tag{10.19}$$

In Example 10.1, we found that, although the discrete time sequence x[n] is real, the discrete frequency sequence X[m] is complex. However, in most applications we are interested in |X[m]|, that is, the magnitude of X[m].

Example 10.3

Use MATLAB to compute the *magnitude* of the frequency components of the following discrete time function. Then, use Excel to display the time and frequency values.

x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]	x[9]	x[10]	x[11]	x[12]	x[13]	x[14]	x[15]
1.0	1.5	2.0	2.3	2.7	3.0	3.4	4.1	4.7	4.2	3.8	3.6	3.2	2.9	2.5	1.8

Solution:

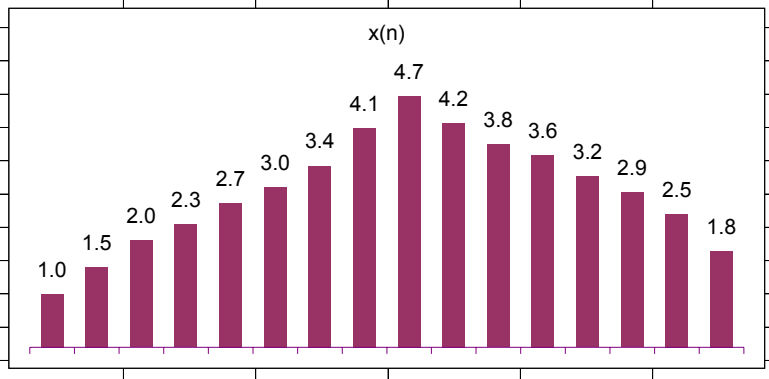
We compute the magnitude of the frequency spectrum with the MATLAB script below.

```
xn=[1 1.5 2 2.3 2.7 3 3.4 4.1 4.7 4.2 3.8 3.6 3.2 2.9 2.5 1.8]; magXm=abs(fft(xn));...
fprintf(' \n'); fprintf('magXm1 = %4.2f \t', magXm(1)); fprintf('magXm2 = %4.2f \t', magXm(2));...
fprintf('magXm3 = %4.2f \t', magXm(3)); fprintf(' \n'); fprintf('magXm4 = %4.2f \t', magXm(4));...
fprintf('magXm5 = %4.2f \t', magXm(5)); fprintf('magXm6 = %4.2f \t', magXm(6)); fprintf(' \n');...
fprintf('magXm7 = %4.2f \t', magXm(7)); fprintf('magXm8 = %4.2f \t', magXm(8));...
fprintf('magXm9 = %4.2f \t', magXm(9)); fprintf(' \n');...
fprintf('magXm10 = %4.2f \t', magXm(10)); fprintf('magXm11 = %4.2f \t', magXm(11)); ...
fprintf('magXm12 = %4.2f \t', magXm(12)); fprintf(' \n');...
fprintf('magXm13 = %4.2f \t', magXm(13)); fprintf('magXm14 = %4.2f \t', magXm(14));...
fprintf('magXm15 = %4.2f \t', magXm(15))
```

```
magXm1 = 46.70   magXm2 = 11.03   magXm3 =  0.42
magXm4 =  2.41   magXm5 =  0.22   magXm6 =  1.19
magXm7 =  0.07   magXm8 =  0.47   magXm9 =  0.10
magXm10 = 0.47   magXm11 = 0.07   magXm12 = 1.19
magXm13 = 0.22   magXm14 = 2.41   magXm15 = 0.42
```

Now, we use Excel to plot the x[n] and |X[m]| values. These are shown in Figure 10.2.

	A	B	C	D
1	**n**	**x(n)**	**m**	**|X(m)|**
2	0	1.0	0	46.70
3	1	1.5	1	11.03
4	2	2.0	2	0.42
5	3	2.3	3	2.41
6	4	2.7	4	0.22
7	5	3.0	5	1.19
8	6	3.4	6	0.07
9	7	4.1	7	0.47
10	8	4.7	8	0.10
11	9	4.2	9	0.47
12	10	3.8	10	0.07
13	11	3.6	11	1.19
14	12	3.2	12	0.22
15	13	2.9	13	2.41
16	14	2.5	14	0.42
17	15	1.8	15	11.03
18				
19				
20				
21				
22				
23				
24				
25				



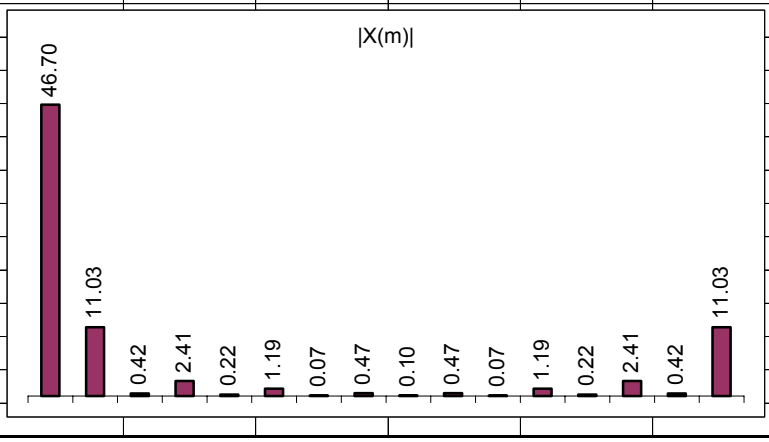


Figure 10.2. Plots of x[n] *and* |X[m]| *values for Example 10.3*

On the plot of |X[m]| in Figure 10.2, the first value X[0] = 46.70 represents the DC component. We observe that after the X[8] = 0.10 value, the magnitude of the frequency components for the range $9 \le m \le 15$, are the mirror image of the components in the range $1 \le m \le 7$. This is not a coincidence; it is a fact that *if* x[n] *is an* N*–point real discrete time function, only* N/2 *of the frequency components of* |X[m]| *are unique*.

Figure 10.3 shows typical discrete time and frequency magnitude waveforms, for a N = 16–point DFT.

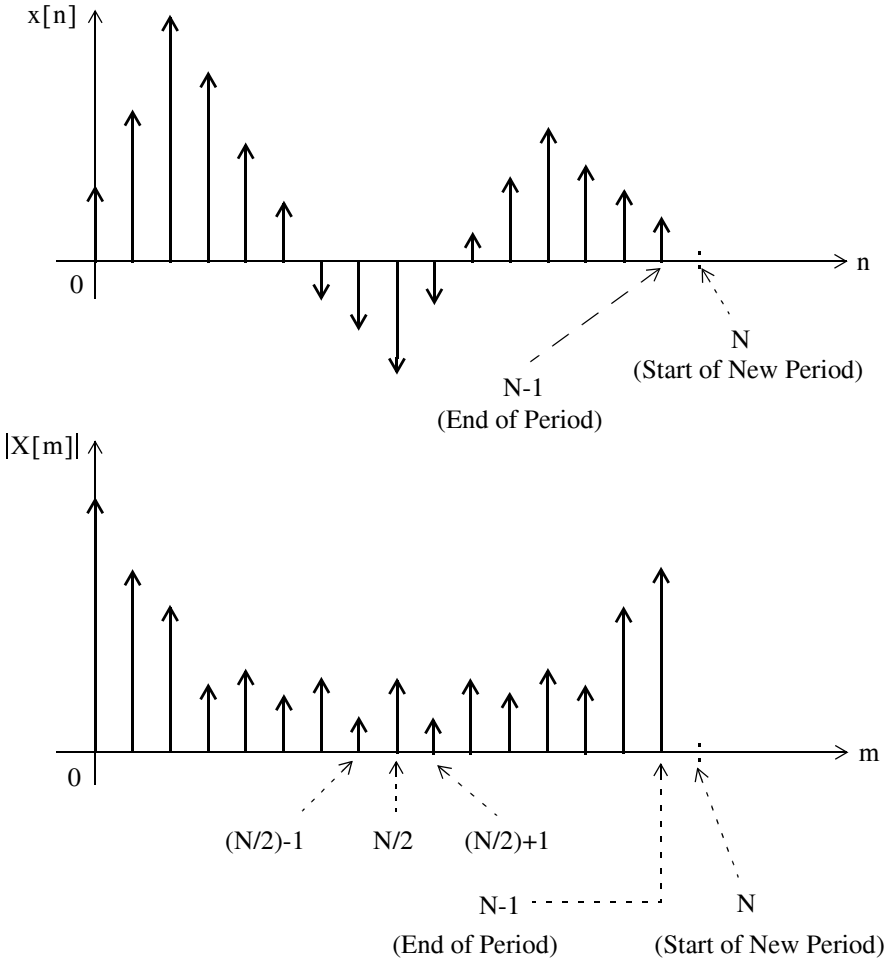


Figure 10.3. x[n] *and* |X[m]| *for a* N = 16 *–point DFT*

Next, we will examine the even and odd properties of the DFT.

10.2 Even and Odd Properties of the DFT

The discrete time and frequency functions are defined as *even* or *odd* in accordance with the following relations:

$$\text{Even Time Function} \qquad f[N-n] = f[n] \tag{10.20}$$

$$\text{Odd Time Function} \qquad f[N-n] = -f[n] \tag{10.21}$$

$$\text{Even Frequency Function} \qquad F[N-m] = F[m] \tag{10.22}$$

$$\text{Odd Frequency Function} \qquad F[N-m] = -F[m] \tag{10.23}$$

In Chapter 8, we developed Table 8-7, Page 8–8, showing the even and odd properties of the Fourier transform. Table 10.2 shows the even and odd properties of the DFT.

TABLE 10.2 Even and Odd Properties of the DFT

Discrete Time Sequence **f[n]**	*Discrete Frequency Sequence* **F[m]**
Real	Complex Real part is Even Imaginary Part is Odd
Real and Even	Real and Even
Real and Odd	Imaginary and Odd
Imaginary	Complex Real part is Odd Imaginary Part is Even
Imaginary and Even	Imaginary and Even
Imaginary and Odd	Real and Odd

The even and odd properties of the DFT shown in Table 10.2 can be proved by methods similar to those that we have used for the continuous Fourier transform. For instance, to prove the first entry we expand

$$X[m] = \sum_{n=0}^{N-1} x[n] W_N^{mn}$$

into its real and imaginary parts using Euler's identity, and we equate these with the real and imaginary parts of $X[m] = X_{re}[m] + jX_{im}[m]$. Now, since the real part contains the cosine, and the imaginary contains the sine function, and $\cos(-m) = \cos m$ while $\sin(-m) = -\sin m$, this entry is proved.

10.3 Common Properties and Theorems of the DFT

The most common properties and theorems of the DFT are presented in Subsections 10.3.1 through 10.3.5 below. For brevity, we will denote the DFT and Inverse DFT as follows:

$$X[m] = \mathfrak{D}\{x[n]\} \tag{10.24}$$

and

$$x[n] = \mathfrak{D}^{-1}\{X[m]\} \tag{10.25}$$

10.3.1 Linearity

$$\boxed{ax_1[n] + bx_2[n] + \ldots \Leftrightarrow aX_1[m] + bX_2[m] + \ldots} \tag{10.26}$$

where $x_1[n] \Leftrightarrow X_1[m]$, $x_2[n] \Leftrightarrow X_2[m]$, and a and b are arbitrary constants.

Proof:

The proof is readily obtained by using the definition of the DFT.

10.3.2 Time Shift

$$\boxed{x[n-k] \Leftrightarrow W_N^{km} X[m]} \tag{10.27}$$

Proof:

By definition,

$$\mathfrak{D}\{x[n]\} = \sum_{n=0}^{N-1} x[n] W_N^{mn}$$

and if $x[n]$ is shifted to the right by k sampled points for $k > 0$, we must change the lower and upper limits of the summation from 0 to k, and from $N-1$ to $N+k-1$ respectively. Then, replacing $x[n]$ with $x[n-k]$ in the definition above, we obtain

$$\mathfrak{D}\{x[n-k]\} = \sum_{n=k}^{N+k-1} x[n-k] W_N^{mn} \tag{10.28}$$

Now, we let $n-k = \mu$; then $n = k+\mu$, and when $n = k$, $\mu = 0$. Therefore, the above relation becomes

$$\mathfrak{D}\{x[\mu]\} = \sum_{\mu=0}^{N-1} x[\mu] W_N^{m(k+\mu)} = W^{km} \sum_{\mu=0}^{N-1} x[\mu] W_N^{\mu m} = W_N^{km} X[\mu] = W_N^{km} X[m] \tag{10.29}$$

We must remember, however, that although the magnitudes of the frequency components are not affected by the shift, a phase shift of $2\pi km/N$ radians is introduced as a result of the time shift. To prove this, let us consider the relation $y[n] = x[n-k]$. Taking the DFT of both sides of this relation, we obtain

$$Y[m] = W_N^{km} X[m] = X[m] e^{-j\frac{2\pi km}{N}} = X[m] \angle \frac{-2\pi km}{N} \tag{10.30}$$

Since both $X[m]$ and $Y[m]$ are complex quantities, they can be expressed in magnitude and phase angle form as

$$X[m] = |X[m]| \angle \theta$$

and

$$Y[m] = |Y[m]| \angle \varphi$$

By substitution of these into (10.30), we obtain

$$|Y[m]|\angle\varphi = |X[m]|\angle\theta\angle\frac{-2\pi km}{N} \tag{10.31}$$

and since $|Y[m]| = |X[m]|$, it follows that

$$\varphi = \theta - \frac{2\pi km}{N} \tag{10.32}$$

10.3.3 Frequency Shift

$$\boxed{W_N^{-km}x[n] \Leftrightarrow X[m-k]} \tag{10.33}$$

Proof:

$$\mathfrak{D}\{W_N^{-kn}x[n]\} = \sum_{n=0}^{N-1} W_N^{-km}x[n]W_N^{mn} = \sum_{n=0}^{N-1} x[n]W_N^{(m-k)n} \tag{10.34}$$

and we observe that the last term on the right side of (10.34) is the same as $\mathfrak{D}\{x[n]\}$ except that *m* is replaced with $m-k$. Therefore,

$$\mathfrak{D}\{W_N^{-km}x[n]\} = X[m-k] \tag{10.35}$$

10.3.4 Time Convolution

$$\boxed{x[n]*h[n] \Leftrightarrow X[m]\cdot H[m]} \tag{10.36}$$

Proof:

Since

$$x[n]*h[n] = \sum_{k=0}^{N-1} x[n]h[n-k]$$

then,

$$\mathfrak{D}\{x[n]*h[n]\} = \sum_{n=0}^{N-1}\left[\sum_{k=0}^{N-1} x[k]h[n-k]\right]W_N^{mn} \tag{10.37}$$

Next, interchanging the order of the indices n and k in the lower limit of the summation, and also changing the range of summation from $N-1$ to $N+k-1$ for the bracketed term on the right side of (10.37), we obtain

$$\mathfrak{D}\{x[n]*h[n]\} = \sum_{k=0}^{N-1} x[k]\left[\sum_{n=k}^{N+k-1} h[n-k]\right]W_N^{kn} \tag{10.38}$$

Now, from the time shifting theorem,

$$\left[\sum_{n=k}^{N+k-1} h[n-k]\right]W_N^{kn} = W_N^{kn}H[m] \tag{10.39}$$

and by substitution into (10.38),

$$\mathfrak{D}\{x[n]*h[n]\} = \left[\sum_{k=0}^{N-1} x[k][W_N^{kn}]\right]H[m] = X[k]\cdot H[m] = X[m]\cdot H[m] \tag{10.40}$$

10.3.5 Frequency Convolution

$$\boxed{x[n]\cdot y[n] \Leftrightarrow \frac{1}{N}\sum_{k=0}^{N-1} X[k]Y[m-k] = \frac{1}{N}X[m]*Y[m]} \tag{10.41}$$

Proof:

The proof is obtained by taking the Inverse DFT, changing the order of the summation, and letting $m-k = \mu$.

10.4 The Sampling Theorem

The *sampling theorem*, also known as *Shannon's Sampling Theorem*, states that *if a continuous time function* $f(t)$ *is band–limited with its highest frequency component less than* W, *then* $f(t)$ *can be completely recovered from its sampled values, if the sampling frequency if equal to or greater than* $2W$.

For example, if we assume that the highest frequency component in a signal is $18\ \text{KHz}$, this signal must be sampled at $2\times 18\ \text{KHz} = 36\ \text{KHz}$ or higher so that it can be completed specified by its sampled values. If the sampled frequency remains the same, i.e., $36\ \text{KHz}$, and the highest frequency of this signal is increased to, say $25\ \text{KHz}$, this signal cannot be recovered by any digital–to–analog converter.

A typical digital signal processing system contains a low–pass analog filter, often called *pre–sampling filter*, to ensure that the highest frequency allowed into the system, will be equal or less the sampling rate so that the signal can be recovered. The highest frequency allowed by the pre–sampling filter is referred to as the *Nyquist frequency,* and it is denoted as f_n.

If a signal is not band–limited, or if the sampling rate is too low, the spectral components of the signal will overlap each another and this condition is called *aliasing*. To avoid aliasing, we must increase the sampling rate.

A discrete time signal may have an infinite length; in this case, it must be limited to a finite interval before it is sampled. We can terminate the signal at the desired finite number of terms, by multiplying it by a *window function*. There are several window functions such as the *rectangular, triangular, Hamming, Hanning, Kaiser,* etc. Window functions are introduced in Appendix E. To obtain a truncated sequence, we multiply an infinite sequence by one of these window functions. However, we must choose a suitable window function; otherwise, the sequence will be terminated abruptly producing the effect of *leakage*. As an example of how leakage can occur, let us review Subsection 8.6.4, Chapter 8, Page 8–30, and the solution of Exercise 8.3, Page 8–50, where the infinite sequence $\cos\omega_0 t$, or $\cos naT$ is multiplied by the window function $A[u_0(t+T)-u_0(t-T)]$ or $A[u_0(n+m)-u_0(n-m)]$. We can see that the spectrum spreads or *leaks* over the neighborhood of $\pm\omega_0$. Selection of an appropriate window function to avoid leakage is beyond the scope of this book, and will not be discussed here.

A third problem that may arise in using the DFT, results from the fact the spectrum of the DFT is not continuous; it is a discrete function where the spectrum consists of integer multiples of the fundamental frequency. It is possible, however, that some significant frequency component lies between two spectral lines and goes undetected. This is called *picket–fence effect* since we can only see discrete values of the spectrum. This problem can be alleviated by adding zeros at the end of the discrete signal, thereby changing the period, which in turn changes the location of the spectral lines.We should remember that the sampling theorem states that the original time sequence can be completely recovered if the sampling frequency is adequate, but does not guarantee that all frequency components will be detected.

To gain a better understanding of the sampling frequency f_s, Nyquist frequency f_n, number of samples N, and the periods in the time and frequency domains, we will adopt the following notations:

N = number of samples in time or frequency period

f_s = sampling frequency = samples per second

T_t = period of a periodic discrete time function

t_t = interval between the N samples in time period T_t

T_f = period of a periodic discrete frequency function

t_f = interval between the N samples in frequency period T_f

These quantities are shown in Figure 10.4. Thus, we have the relations

$$t_t = \frac{T_t}{N} \qquad f_s = \frac{1}{t_t} \qquad t_f = \frac{T_f}{N} \qquad t_t = \frac{1}{T_f} \qquad t_f = \frac{1}{T_t} \tag{10.42}$$

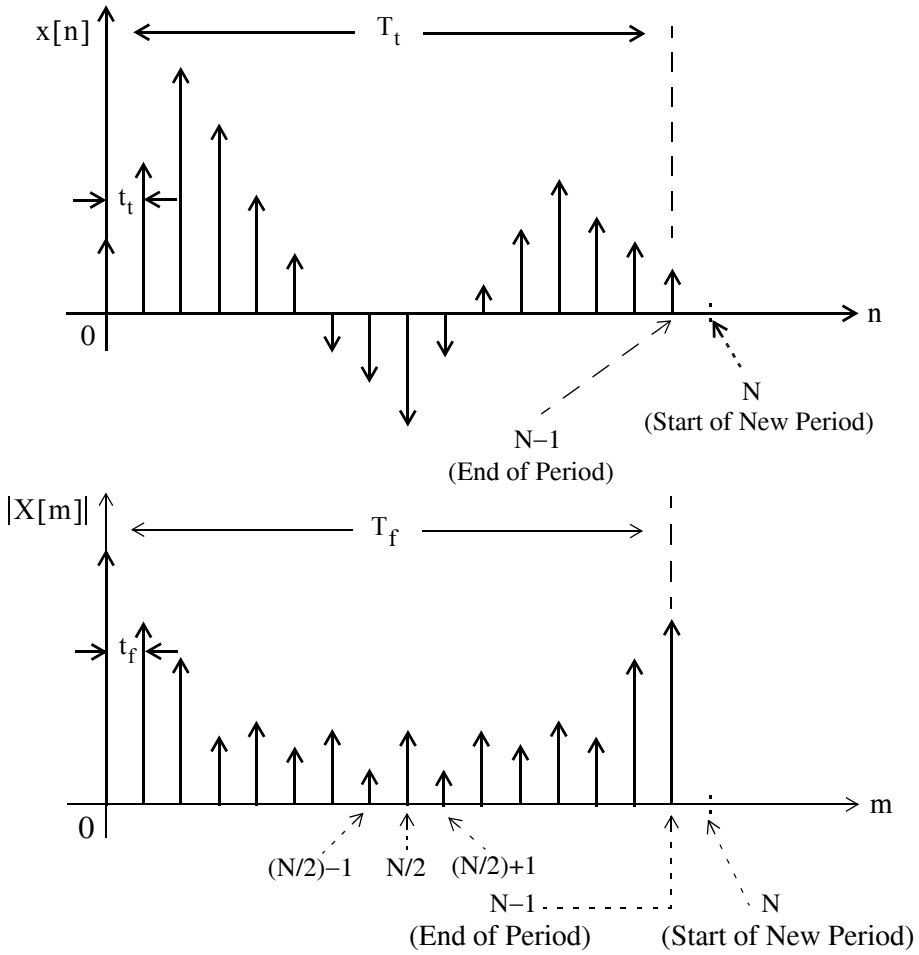


Figure 10.4. Intervals between samples and periods in discrete time and frequency domains

Example 10.4

The period of a periodic discrete time function is 0.125 millisecond, and it is sampled at 1024 equally spaced points. It is assumed that with this number of samples, the sampling theorem is satisfied and thus there will be no aliasing.

a. Compute the period of the frequency spectrum in KHz.

b. Compute the interval between frequency components in KHz.

c. Compute the sampling frequency f_s

d. Compute the Nyquist frequency f_n

Solution:

For this example, $T_t = 0.125$ ms and $N = 1024$ points. Therefore, the time between successive time components is

$$t_t = \frac{T_t}{N} = \frac{0.125 \times 10^{-3}}{1024} = 0.122\ \mu s$$

Then,

a. the period T_f of the frequency spectrum is

$$T_f = \frac{1}{t_t} = \frac{1}{0.122 \times 10^{-6}} = 8192\ \text{kHz} \approx 8.2\ \text{MHz}$$

b. the interval t_f between frequency components is

$$t_f = \frac{T_f}{N} = \frac{8192\ \text{kHz}}{1024} = 8\ \text{kHz}$$

c. the sampling frequency f_s is

$$f_s = \frac{1}{t_t} = \frac{1}{0.122 \times 10^{-6}} = 8.2\ \text{MHz}$$

d. the Nyquist frequency must be equal or less than half the sampling frequency, that is,

$$f_n \le \frac{1}{2} f_s \le \frac{1}{2} \times 8.2\ \text{MHz} \le 4.1\ \text{MHz}$$

10.5 Number of Operations Required to Compute the DFT

Let us consider a signal whose highest (Nyquist) frequency is 18 KHz, the sampling frequency is 50 KHz, and 1024 samples are taken, i.e., $N = 1024$. The time required to compute the entire DFT would be

$$t = \frac{1024\ \text{samples}}{50 \times 10^3\ \text{samples per second}} = 20.48\ \text{ms} \tag{10.43}$$

To compute the number of operations required to complete this task, let us expand the N–point DFT defined as

$$X[m] = \sum_{n=0}^{N-1} x[n] W_N^{mn} \tag{10.44}$$

Then,

$$\begin{aligned}
X[0] &= x[0]W_N^0 + x[1]W_N^0 + x[2]W_N^0 + \ldots + x[N-1]W_N^0 \\
X[1] &= x[0]W_N^0 + x[1]W_N^1 + x[2]W_N^2 + \ldots + x[N-1]W_N^{N-1} \\
X[2] &= x[0]W_N^0 + x[1]W_N^2 + x[2]W_N^4 + \ldots + x[N-1]W_N^{2(N-1)} \\
&\ldots \\
X[N-1] &= x[0]W_N^0 + x[1]W_N^{N-1} + x[2]W_N^{2(N-1)} + \ldots + x[N-1]W_N^{(N-1)^2}
\end{aligned} \tag{10.45}$$

and it is worth remembering that

$$W_N^0 = e^{\left(-j\frac{2\pi}{N}\right)(0)} = 1 \tag{10.46}$$

Since W_N^i is a complex number, the computation of any frequency component $X[k]$, requires N complex multiplications and N complex additions, that is, $2N$ complex arithmetic operations are required to compute any frequency component of $X[k]$. If we assume that $x[n]$ is real, then only $N/2$ of the $||X[m]|$ components are unique. Therefore, we would require $2N \times N/2 = N^2$ complex operations to compute the entire frequency spectrum. Thus, for an $N = 1024$–point DFT, such as the one with 18 KHz signal, we would require $N^2 = 1024^2 = 1048576$ complex operations, and these would have to be completed within 20.48 ms as we found in (10.43). Although the means of doing this task may be possible with today's technology, it seems impractical.

Fortunately, many of the W_N terms in (10.45) are unity. Moreover, because of some symmetry properties, the number of complex operations can be reduced considerably. This is possible with the algorithm known as FFT (Fast Fourier Transform) that was developed by Cooley and Tukey, and it is very well documented. This algorithm is the subject of the next section.

10.6 The Fast Fourier Transform (FFT)

In this section, we will be making extensive use of the complex rotating vector

$$W_N = e^{-\frac{j2\pi}{N}} \tag{10.47}$$

and the additional properties of W_N in (10.48) below.

$$
\begin{aligned}
W_N^N &= e^{-\left(\frac{j2\pi}{N}\right)N} = e^{-j2\pi} = 1 \\
W_N^{N/2} &= e^{-\left(\frac{j2\pi}{N}\right)\frac{N}{2}} = e^{-j\pi} = -1 \\
W_N^{N/4} &= e^{-\left(\frac{j2\pi}{N}\right)\frac{N}{4}} = e^{-j\pi/2} = -j \\
W_N^{3N/4} &= e^{-\left(\frac{j2\pi}{N}\right)\frac{3N}{4}} = e^{-j3\pi/2} = j \\
W_N^{kN} &= e^{-\left(\frac{j2\pi}{N}\right)kN} = e^{-j2k\pi} = 1 \quad k = 0, 1, 2, \ldots \\
W_N^{kN+r} &= e^{-\left(\frac{j2\pi}{N}\right)(kN+r)} = e^{-\left(\frac{j2\pi}{N}\right)kN} e^{-\left(\frac{j2\pi}{N}\right)r} = 1 \cdot W_N^r = W_N^r \\
W_{2N}^k &= e^{-\left(\frac{j2\pi}{2N}\right)k} = e^{-\left(\frac{j2\pi}{N}\right)\frac{k}{2}} = W_N^{k/2}
\end{aligned}
\tag{10.48}
$$

We rewrite the array of (10.45) in matrix form as shown in (10.49) below.

$$
\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \cdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ W_N^0 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix} \cdot \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \cdots \\ x[N-1] \end{bmatrix} \tag{10.49}
$$

This is a complex Vandermonde matrix and it is expressed in a more compact form as

$$
X[m] = \left[W_N\right] \cdot x[n] \tag{10.50}
$$

The algorithm that was developed by *Cooley* and *Tukey*, is based on matrix decomposition methods, where the matrix W_N in (10.50) is factored into L smaller matrices, that is,

$$
\left[W_N\right] = \left[W_1\right] \cdot \left[W_2\right] \cdot \ldots \cdot \left[W_L\right] \tag{10.51}
$$

where L is chosen as $L = log_2 N$ or $N = 2^L$.

Each row of the matrices on the right side of (10.51) contains only two non-zero terms, unity and W_N^k. Then, the vector $X[m]$ is obtained from

$$X[m] = \left[W_1\right] \cdot \left[W_2\right] \cdot \ldots \cdot \left[W_L\right] \cdot x[n] \tag{10.52}$$

The FFT computation begins with matrix $\left[W_L\right]$ in (10.52). This matrix operates on vector $x[n]$ producing a new vector, and each component of this new vector, is obtained by one multiplication and one addition. This is because there are only two non–zero elements on a given row, and one of these elements is unity. Since there are N components on $x[n]$, there will be N complex additions and N complex multiplications. This new vector is then operated on by the $[W_{L-1}]$ matrix, then on $[W_{L-2}]$, and so on, until the entire computation is completed. It appears that the entire computation would require $NL = N\log_2 N$ complex multiplications, and also $N\log_2 N$ additions for a total of $2N\log_2 N$ arithmetic operations. However, since $W_N^0 = 1$, $W_N^{N/2} = -1$, and other reductions, it is estimated that only about half of these, that is, $N\log_2 N$ total arithmetic operations are required by the FFT versus the N^2 computations required by the DFT.

Under those assumptions, we construct Table 10.3 to compare the percentage of computations achieved by the use of FFT versus the DFT.

TABLE 10.3 DFT and FFT Computations

	DFT	FFT	FFT/DFT
N	N^2	$N\log_2 N$	%
8	64	24	37.5
16	256	64	25
32	1024	160	15.6
64	4096	384	9.4
128	16384	896	5.5
256	65536	2048	3.1
512	262144	4608	1.8
1024	1048576	10240	1
2048	4194304	22528	0.5

A plethora of FFT algorithms has been developed and published. They are divided into two main categories:

Category I

a. *Decimation in Time*

b. *Decimation in Frequency*

Category II

a. *In–Place*

b. *Natural Input–Output (Double–Memory Technique)*

To define Category I, we need to refer to the DFT and Inverse DFT definitions. They are repeated below for convenience.

$$X[m] = \sum_{n=0}^{N-1} x[n]W_N^{mn} \tag{10.53}$$

and

$$x[n] = \frac{1}{N}\sum_{n=0}^{N-1} X[m]W_N^{-mn} \tag{10.54}$$

We observe that (10.53) and (10.54) differ only by the factor $1/N$, and the replacement of W_N with W_N^{-1}. If the DFT algorithm is developed in terms of the direct DFT of (10.53), it is referred to as *decimation in time,* and if it is developed in terms of the Inverse DFT of (10.54), it is referred to as *decimation in frequency*. In the latter case, the vector

$$W_N = e^{-j\frac{2\pi}{N}}$$

is replaced by its complex conjugate

$$W_N^{-1} = e^{j\frac{2\pi}{N}}$$

that is, the sine terms are reversed in sign, and the multiplication by the factor $1/N$ can be done either at the input or output.

The Category II algorithm schemes are described in the Table 10.4 along with their advantages and disadvantages.

TABLE 10.4 In-Place and Natural Input–Output algorithms

Category II	Description	Advantages	Disadvantages
In–Place	The process where the result of a computation of a new vector is stored in the same memory location as the result of the previous computation	Eliminates the need for intermediate storage and memory require-ments	The output appears in an unnatural order and must be re–ordered.
Natural Input–Output (Double Memory)	The process where the output appears in same (natural) order as the input	No re–ordering is required	Requires more internal memory to pre-serve the natural order

Now, we will explain how the unnatural order occurs and how it can be re–ordered.

Consider the discrete time sequence $f[n]$; its DFT is found from

$$F[m] = \sum_{n=0}^{N-1} f[n]W_N^{mn} \tag{10.55}$$

We assume that N is a power of 2 and thus, it is divisible by 2. Then, we can decompose the sequence $f[n]$ into two subsequences, $f_{even}[n]$ which contains the even components, and $f_{odd}[n]$ which contains the odd components. In other words, we choose these as

$$f_{even}[n] = f[2n]$$

and

$$f_{odd}[n] = f[2n+1]$$

for $n = 0, 1, 2, \ldots N/2 - 1$

Each of these subsequences has a length of $N/2$ and thus, their DFTs are, respectively,

$$F_{even}[m] = \sum_{n=0}^{N/2-1} f_{even}[n]W_{N/2}^{mn} = \sum_{n=0}^{N/2-1} f[2n]W_{N/2}^{mn} \tag{10.56}$$

and

$$F_{odd}[m] = \sum_{n=0}^{N/2-1} f_{odd}[n]W_{N/2}^{mn} = \sum_{n=0}^{N/2-1} f[2n+1]W_{N/2}^{mn} \tag{10.57}$$

where

$$W_{N/2} = e^{-j\frac{2\pi}{N/2}} = e^{\left(-j\frac{2\pi}{N}\right)2} = W_N^2 \tag{10.58}$$

For an 8–point DFT, $N = 8$. Expanding (10.55) for $n = 1, 2, 3, \ldots, 7$ we obtain

$$\begin{aligned} F[m] &= \sum_{n=0}^{7} f[n]W_N^{mn} \\ &= f[0] + f[1]W_N^{m} + f[2]W_N^{2m} + f[3]W_N^{3m} \\ &\quad + f[4]W_N^{4m} + f[5]W_N^{5m} + f[6]W_N^{6m} + f[7]W_N^{7m} \end{aligned} \tag{10.59}$$

Expanding (10.56) for $n = 0, 1, 2,$ and 3 and recalling that $W_N^0 = 1$, we obtain

$$\begin{aligned} F_{even}[m] &= \sum_{n=0}^{3} f[2n]W_N^{2mn} \\ &= f[0]W_N^{0} + f[2]W_N^{2m} + f[4]W_N^{4m} + f[6]W_N^{6m} \\ &= f[0] + f[2]W_N^{2m} + f[4]W_N^{4m} + f[6]W_N^{6m} \end{aligned} \tag{10.60}$$

Expanding also (10.57) for $n = 0, 1, 2,$ and 3 and using $W_N^0 = 1$, we obtain

$$\begin{aligned} F_{odd}[m] &= \sum_{n=0}^{3} f[2n+1]W_N^{2mn} = f[1]W_N^{0} + f[3]W_N^{2m} + f[5]W_N^{4m} + f[7]W_N^{6m} \\ &= f[1] + f[3]W_N^{2m} + f[5]W_N^{4m} + f[7]W_N^{6m} \end{aligned} \tag{10.61}$$

The vector W_N is the same in (10.59), (10.60) and (10.61), and $N = 8$. Then,

$$W_N = W_8 = e^{-j\frac{2\pi}{N}} = e^{-j\frac{2\pi}{8}}$$

Multiplying both sides of (10.61) by W_N^m, we obtain

$$W_N^m F_{odd}[m] = f[1]W_N^m + f[3]W_N^{3m} + f[5]W_N^{5m} + f[7]W_N^{7m} \tag{10.62}$$

and from (10.59), (10.60) and (10.62), we observe that

$$F[m] = F_{even}[m] + W_N^m F_{odd}[m] \tag{10.63}$$

for $n = 1, 2, 3, \ldots, 7$.

Continuing the process, we decompose $\{f[0], f[2], f[4], \text{and } f[6]\}$ into $\{f[0], f[4]\}$ and $\{f[2], f[6]\}$. These are sequences of length $N/4 = 2$.

Denoting their DFTs as $F_{even1}[m]$ and $F_{even2}[m]$, and using the relation

$$W_{N/4} = e^{-j\frac{2\pi}{N/4}} = e^{\left(-j\frac{2\pi}{N}\right)4} = W_N^4 \tag{10.64}$$

for $n = 0, 1$, we obtain

$$F_{even1}[m] = f[0] + f[4]W_N^{4m} \tag{10.65}$$

and

$$F_{even2}[m] = f[2] + f[6]W_N^{4m} \tag{10.66}$$

The sequences of (10.65) and (10.66) cannot be decomposed further. They justify the statement made earlier, that each computation produces a vector where each component of this vector, for $n = 1, 2, 3, \ldots, 7$ is obtained by one multiplication and one addition. This is often referred to as a *butterfly operation*.

Substitution of (10.65) and (10.66) into (10.60), yields

$$F_{even}[m] = F_{even1}[m] + W_N^{2m} F_{even2}[m] \tag{10.67}$$

Likewise, $F_{odd}[m]$ can be decomposed into DFTs of length 2; then, $F[m]$ can be computed from

$$F(m) = F_{even}(m) + W_N^m F_{odd}(m) \quad m = 0, 1, 2, \ldots, 7 \tag{10.68}$$

for $N = 8$. Of course, this procedure can be extended for any N that is divisible by 2.

Figure 10.5 shows the *signal flow graph* of a decimation in time, in–place FFT algorithm for $N = 8$, where the input is shuffled in accordance with the above procedure. The subscript N in W has been omitted for clarity.

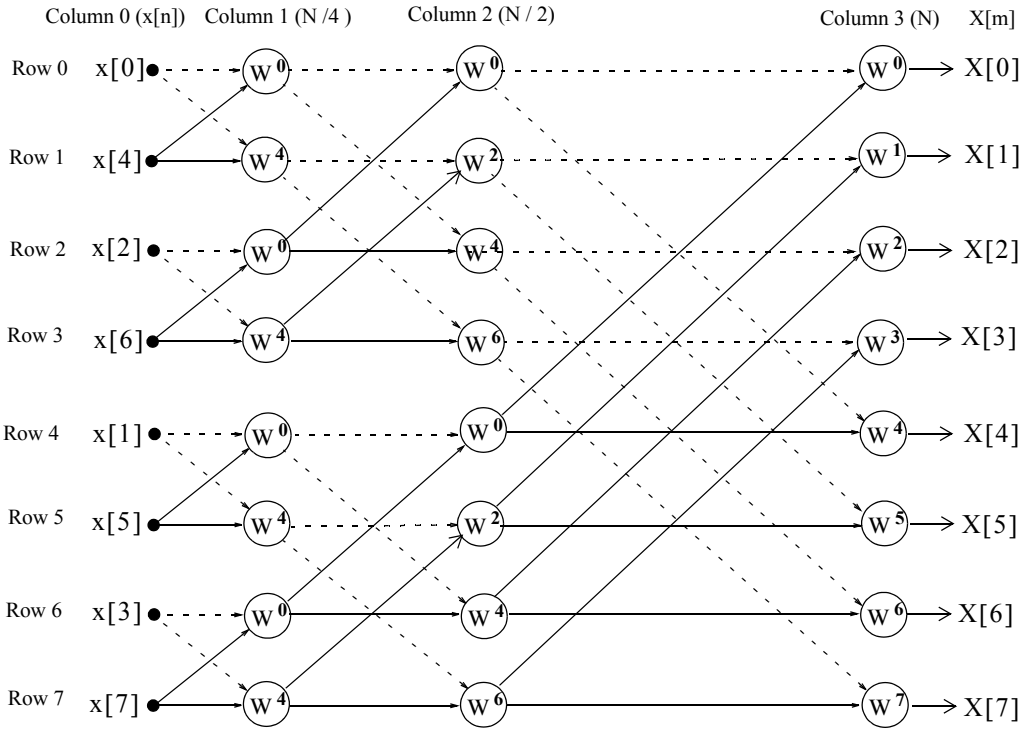


Figure 10.5. Signal flow graph of a decimation in time, in-place FFT algorithm, for $N = 8$

In the signal flow graph of Figure 10.5, the input x[n] appears in Column 0. The N/4, N/2, and N–point FFTs are in Columns 1, 2, and 3 respectively.

In simplified form, the output of the node associated with row R and column C, indicated as Y(R, C), is found from

$$Y[R, C] = \underbrace{Y[R_i, C-1]}_{\text{Dash line}} + \underbrace{W^m Y[R_j, C-1]}_{\text{Solid line}} \tag{10.69}$$

where $0 \le R \le 7$, $0 \le C \le 3$, and the exponent m in W^m is indicated on the signal flow graph.

The binary input words and the bit–reversed words applicable to the this signal flow graph, are shown in Table 10.5.

TABLE 10.5 Binary words for the signal flow graph of Figure 10.5

n	Binary Word	Reversed-Bit Word	Input Order
0	000	000	*x*[0]
1	001	100	*x*[4]
2	010	010	*x*[2]
3	011	110	*x*[6]
4	100	001	*x*[1]
5	101	101	*x*[5]
6	110	011	*x*[3]
7	111	111	*x*[7]

We will illustrate the use of (10.69) with the following example.

Example 10.5

Using (10.69) with the signal flow graph of Figure 10.5, compute the spectral component X[3] in terms of the inputs x[i] and vectors W^j. Then, verify that the result is the same as that obtained by direct application of the DFT.

Solution:

The N/4–point FFT appears in Column 1. Using (10.69) we obtain:

$$
\begin{aligned}
Y[0,1] &= x[0] + W^{0}x[4] = Y[0,0] + W^{0}Y[1,0] \\
Y[1,1] &= x[0] + W^{4}x[4] = Y[0,0] + W^{4}Y[1,0] \\
Y[2,1] &= x[2] + W^{0}x[6] = Y[2,0] + W^{0}Y[3,0] \\
Y[3,1] &= x[2] + W^{4}x[6] = Y[2,0] + W^{4}Y[3,0] \\
Y[4,1] &= x[1] + W^{0}x[5] = Y[4,0] + W^{0}Y[5,0] \\
Y[5,1] &= x[1] + W^{4}x[5] = Y[4,0] + W^{4}Y[5,0] \\
Y[6,1] &= x[3] + W^{0}x[7] = Y[6,0] + W^{0}Y[7,0] \\
Y[7,1] &= x[3] + W^{0}x[7] = Y[6,0] + W^{4}Y[7,0]
\end{aligned}
\tag{10.70}
$$

The $N/2$–point FFT appears in Column 2. Using (10.69) we obtain:

$$
\begin{aligned}
Y[0,2] &= Y[0,1] + W^{0}Y[2,1] \\
Y[1,2] &= Y[1,1] + W^{2}Y[3,1] \\
Y[2,2] &= Y[0,1] + W^{4}Y[2,1] \\
Y[3,2] &= Y[1,1] + W^{6}Y[3,1] \\
Y[4,2] &= Y[4,1] + W^{0}Y[6,1] \\
Y[5,2] &= Y[5,1] + W^{2}Y[7,1] \\
Y[6,2] &= Y[4,1] + W^{4}Y[6,1] \\
Y[7,2] &= Y[5,1] + W^{6}Y[7,1]
\end{aligned}
\tag{10.71}
$$

The N–point FFT appears at the outputs of Column 3, where

$$
\begin{aligned}
Y[0,3] &= Y[0,2] + W^{0}Y[4,2] \\
Y[1,3] &= Y[1,2] + W^{1}Y[5,2] \\
Y[2,3] &= Y[2,2] + W^{2}Y[6,2] \\
Y[3,3] &= Y[3,2] + W^{3}Y[7,2] \\
Y[4,3] &= Y[0,2] + W^{4}Y[4,2] \\
Y[5,3] &= Y[1,2] + W^{5}Y[5,2] \\
Y[6,3] &= Y[2,2] + W^{6}Y[6,2] \\
Y[7,3] &= Y[3,2] + W^{7}Y[7,2]
\end{aligned}
\tag{10.72}
$$

With the equations of (10.70), (10.71), and (10.72), we can determine any of the 8 spectral components. Our example calls for X [3] ; then, with reference to the signal flow chart of Figure 10.5 and the fourth of the equations in (10.72),

$$X[3] = Y[3,3] = Y[3,2] + W^{3}Y[7,2] \tag{10.73}$$

Also, from (10.71)

$$Y[3,2] = Y[1,1] + W^{6}Y[3,1] \tag{10.74}$$

and

$$Y[7,2] = Y[5,1] + W^{6}Y[7,1] \tag{10.75}$$

Finally, from (10.70)

$$\begin{aligned} Y[1,1] &= Y[0,0] + W^{4}Y[1,0] \\ Y[3,1] &= Y[2,0] + W^{4}Y[3,0] \\ Y[5,1] &= Y[4,0] + W^{4}Y[5,0] \\ Y[7,1] &= Y[6,0] + W^{4}Y[7,0] \end{aligned} \tag{10.76}$$

and making these substitutions into (10.73), we obtain

$$\begin{aligned} Y[3,3] &= Y[3,2] + W^{3}Y[7,2] \\ &= Y[1,1] + W^{6}Y[3,1] + W^{3}\{Y[5,1] + W^{6}Y[7,1]\} \\ &= Y[0,0] + W^{4}Y[1,0] + W^{6}\{Y[2,0] + W^{4}Y[3,0]\} \\ &\quad + W^{3}\{(Y[4,0] + W^{4}Y[5,0]) + W^{6}(Y[6,0] + W^{4}Y[7,0])\} \\ &= Y[0,0] + W^{4}Y[1,0] + W^{6}Y[2,0] + W^{10}Y[3,0] \\ &\quad + W^{3}Y[4,0] + W^{7}Y[5,0] + W^{9}Y[6,0] + W^{13}Y[7,0] \end{aligned}$$

Rearranging in ascending powers of W , we obtain

$$\begin{aligned} Y[3,3] &= Y[0,0] + W^{3}Y[4,0] + W^{4}Y[1,0] + W^{6}Y[2,0] \\ &\quad + W^{7}Y[5,0] + W^{9}Y[6,0] + W^{10}Y[3,0] + W^{13}Y[7,0] \end{aligned} \tag{10.77}$$

From the signal flow graph of Figure 10.5, we observe that

$$\begin{aligned} Y[0,0] &= x[0] \qquad Y[4,0] = x[1] \\ Y[1,0] &= x[4] \qquad Y[2,0] = x[2] \\ Y[5,0] &= x[5] \qquad Y[6,0] = x[3] \\ Y[3,0] &= x[6] \qquad Y[7,0] = x[7] \end{aligned}$$

and

$$Y[3,3] = X[3]$$

Therefore, we express (10.77) as

$$\begin{aligned} X[3] &= x[0] + x[1]W^{3} + x[4]W^{4} + x[2]W^{6} \\ &\quad + x[5]W^{7} + x[3]W^{9} + x[6]W^{10} + x[7]W^{13} \\ &= x[0] + x[1]W^{3} + x[2]W^{6} + x[3]W^{9} \\ &\quad + x[4]W^{4} + x[5]W^{7} + x[6]W^{10} + x[7]W^{13} \end{aligned} \tag{10.78}$$

We will verify that this expression is correct by the use of the direct DFT of (10.17), that is,

$$X[m] = \sum_{n=0}^{N-1} x(n)W_N^{mn}$$

For $m = 3$ and $n = 0, 1, 2, \ldots, 7$, we obtain

$$\begin{aligned} X[3] &= x[0] + x[1]W^{3} + x[2]W^{6} + x[3]W^{9} \\ &\quad + x[4]W^{12} + x[5]W^{15} + x[6]W^{18} + x[7]W^{21} \end{aligned} \tag{10.79}$$

and from (10.48),

$$W^{kN+r} = W^{r}$$

Now, with $N = 8$, and $k = 1$, we obtain $kN = 8$. Then,

$$W^{12} = W^{4}, \quad W^{15} = W^{7}, \quad W^{18} = W^{10} \quad \text{and} \quad W^{21} = W^{13}$$

and by substitution into (10.79),

$$\begin{aligned} X[3] &= x[0] + x[1]W^{3} + x[2]W^{6} + x[3]W^{9} \\ &\quad + x[4]W^{4} + x[5]W^{7} + x[6]W^{10} + x[7]W^{13} \end{aligned} \tag{10.80}$$

We observe that (10.80) is the same as (10.78).

The signal flow graph of Figure 10.5, represents a shuffled input, natural output algorithm. Others are natural input, natural output, or natural input, shuffled output. These combinations occur in both decimation in time, and decimation in frequency algorithms.

10.7 Summary

- The N–point DFT is defined as

$$X[m] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{mn}{N}}$$

where N represents the number of points that are equally spaced in the interval 0 to 2π on the unit circle of the z-plane, and $m = 0, 1, 2, \ldots, N-1$.

- The N–point Inverse DFT is defined as

$$x[n] = \frac{1}{N}\sum_{m=0}^{N-1} X[m]e^{j2\pi\frac{mn}{N}}$$

for $n = 0, 1, 2, \ldots, N-1$.

- In general, the discrete frequency transform $X[m]$ is complex, and it is expressed as

$$X[m] = Re\{X[m]\} + Im\{X[m]\}$$

The real part $Re\{X[m]\}$ can be computed from

$$Re\{X[m]\} = x[0] + \sum_{n=1}^{N-1} x[n]\cos\frac{2\pi mn}{N} \quad \text{for} \quad m = 0, 1, 2, \ldots, N-1$$

and the imaginary part from

$$Im\{X[m]\} = -\sum_{n=1}^{N-1} x[n]\sin\frac{2\pi mn}{N} \quad \text{for} \quad m = 0, 1, 2, \ldots, N-1$$

- We can use the **fft(x)** function to compute the DFT, and the **ifft(x)** function to compute the Inverse DFT.

- The term $e^{-(j2\pi)/N}$ is a rotating vector, where the range $0 \le \theta \le 2\pi$ is divided into $360/N$ equal segments and it is denoted as represent it as W_N, that is,

$$W_N = e^{-\frac{j2\pi}{N}}$$

and consequently

$$W_N^{-1} = e^{\frac{j2\pi}{N}}$$

Accordingly, the DFT pair is normally denoted as

$$X[m] = \sum_{n=0}^{N-1} x(n) W_N^{mn}$$

and

$$x[n] = \frac{1}{N} \sum_{n=0}^{N-1} X(m) W_N^{-mn}$$

- The correspondence between $x[n]$ and $X[m]$ is denoted as

$$x[n] \Leftrightarrow X[m]$$

- If $x[n]$ is an N–point real discrete time function, only $N/2$ of the frequency components of $|X[m]|$ are unique.
- The discrete time and frequency functions are defined as even or odd, in accordance with the relations

Even Time Function	$f[N-n] = f[n]$
Odd Time Function	$f[N-n] = -f[n]$
Even Frequency Function	$F[N-m] = F[m]$
Odd Frequency Function	$F[N-m] = -F[m]$

- The even and odd properties of the DFT are similar to those of the continuous Fourier transform and are listed in Table 10.2.
- The linearity property of the DFT states that

$$ax_1[n] + bx_2[n] + \ldots \Leftrightarrow aX_1[m] + bX_2[m] + \ldots$$

- The time shift property of the DFT states that

$$x[n-k] \Leftrightarrow W_N^{km} X[m]$$

- The frequency shift property of the DFT states that

$$W_N^{-km} x[n] \Leftrightarrow X[m-k]$$

- The time convolution property of the DFT states that

$$x[n]*h[n] \Leftrightarrow X[m] \cdot H[m]$$

- The frequency convolution property of the DFT states that

$$x[n] \cdot y[n] \Leftrightarrow \frac{1}{N} \sum_{k=0}^{N-1} X[k]Y[m-k] = \frac{1}{N} X[m]*Y[m]$$

- The sampling theorem, also known as Shannon's Sampling Theorem, states that if a continuous time function $f(t)$ is band-limited with its highest frequency component less than W, then $f(t)$ can be completely recovered from its sampled values, if the sampling frequency if equal to or greater than $2W$.
- A typical digital signal processing system contains a low-pass analog filter, often called pre-sampling filter, to ensure that the highest frequency allowed into the system, will be equal or less the sampling rate so that the signal can be recovered. The highest frequency allowed by the pre-sampling filter is referred to as the Nyquist frequency, and it is denoted as f_n.
- If a signal is not band-limited, or if the sampling rate is too low, the spectral components of the signal will overlap each another and this condition is called aliasing.
- If a discrete time signal has an infinite length, we can terminate the signal at a desired finite number of terms, by multiplying it by a window function. However, we must choose a suitable window function; otherwise, the sequence will be terminated abruptly producing the effect of leakage
- If in a discrete time signal some significant frequency component that lies between two spectral lines and goes undetected, the picket-fence effect is produced. This effect can be alleviated by adding zeros at the end of the discrete signal, thereby changing the period, which in turn changes the location of the spectral lines.
- The number of operations required to compute the DFT can be significantly reduced by the FFT algorithm.
- The Category I FFT algorithms are classified either as decimation it time or decimation in frequency. Decimation in time implies that DFT algorithm is developed in terms of the direct DFT, whereas decimation in frequency implies that the DFT is developed in terms of the Inverse DFT.
- The Category II FFT algorithms are classified either as in-place or natural input–output. In-place refers to the process where the result of a computation of a new vector is stored in the same memory location as the result of the previous computation. Natural input–output refers to the process where the output appears in same (natural) order as the input.
- The FFT algorithms are usually shown in a signal flow graph. In some signal flow graphs the input is shuffled and the output is natural, and in others the input is natural and the output is shuffled. These combinations occur in both decimation in time, and decimation in frequency algorithms.

10.8 Exercises

1. Compute the DFT of the sequence $x[0] = x[1] = 1,\ x[2] = x[3] = -1$

2. A square waveform is represented by the discrete time sequence

$$x[0] = x[1] = x[2] = x[3] = 1 \text{ and } x[4] = x[5] = x[6] = x[7] = -1$$

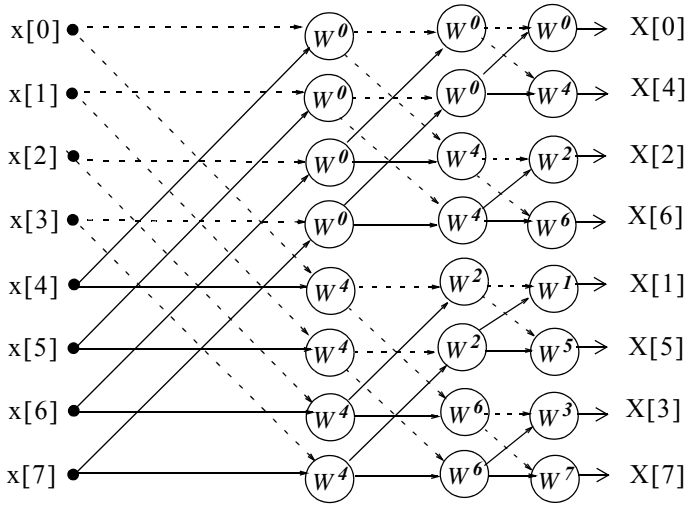
Use MATLAB to compute and plot the magnitude $|X[m]|$ of this sequence.

3. Prove that

a. $x[n]\cos\frac{2\pi kn}{N} \Leftrightarrow \frac{1}{2}\{X[m-k] + X[m+k]\}$

b. $x[n]\sin\frac{2\pi kn}{N} \Leftrightarrow \frac{1}{j2}\{X[m-k] + X[m+k]\}$

4. The signal flow below is a decimation in time, natural–input, shuffled–output type FFT algorithm. Using this graph and relation (10.69), compute the frequency component $X[3]$. Verify that this is the same as that found in Example 10.5.



5. The signal flow graph below is a decimation in frequency, natural input, shuffled output type *FFT* algorithm. There are two equations that relate successive columns. The first is

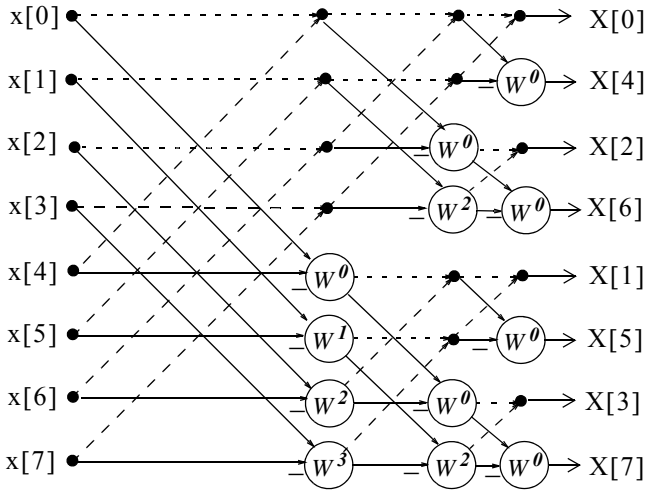
$$Y_{dash}(R, C) = Y_{dash}(R_i, C-1) + Y_{dash}(R_j, C-1)$$

and it is used with the nodes where two dashed lines terminate on them.

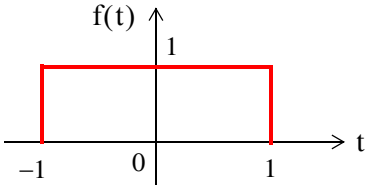
The second equation is

$$Y_{sol}(R, C) = W^m[Y_{sol}(R_i, C-1) - Y_{sol}(R_j, C-1)]$$

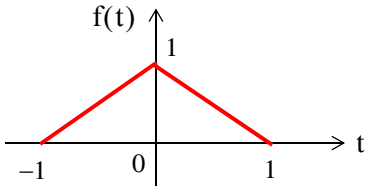
and it is used with the nodes where two solid lines terminate on them. The number inside the circles denote the power of W_N, and the minus (–) sign below serves as a reminder that the bracketed term of the second equation involves a subtraction. Using this graph and the above equations, compute the frequency component $X[3]$. Verify that this is the same as in Example 10.5.



6. Plot the Fourier transform of the rectangular pulse shown below, using the MATLAB **fft** function. Then, use the **ifft** function to verify that the inverse transformation produces the rectangular pulse.



7. Plot the Fourier transform of the triangular pulse shown below using the MATLAB **fft** function. Then, use the **ifft** function to verify that the inverse transformation produces the rectangular pulse.



10.9 Solutions to End–of–Chapter Exercises

1.

$$F[m] = \sum_{n=0}^{N-1} f(n)W_N^{mn}$$

where $N = 4$ and $f[0] = f[1] = 1$, $f[2] = f[3] = -1$. Then,

$$F[m] = \sum_{n=0}^{3} f(n)W_4^{mn} = f[0]W_4^{m[0]} + f[1]W_4^{m[1]} + f[2]W_4^{m[2]} + f[3]W_4^{m[3]}$$

for $m = 0, 1, 2,$ and 3

$m = 0$:

$$\begin{aligned} F(0) &= (1)\cdot(e^0) + (1)\cdot(e^0) + (-1)\cdot(e^0) + (-1)\cdot(e^0) \\ &= (1)\cdot(1) + (1)\cdot(1) + (-1)\cdot(1) + (-1)\cdot(1) = 0 \end{aligned}$$

$m = 1$:

$$\begin{aligned} F(1) &= (1)\cdot(e^0) + (1)\cdot(e^{(-j2\pi/4)\times 1}) + (-1)\cdot(e^{(-j2\pi/4)\times 2}) + (-1)\cdot(e^{(-j2\pi/4)\times 3}) \\ &= 1 + \cos\frac{\pi}{2} - j\sin\frac{\pi}{2} - \cos\pi + j\sin\pi - \cos\frac{3\pi}{2} + j\sin\frac{3\pi}{2} \\ &= 1 + 0 - j + 1 + 0 - 0 - j = 2 - j2 \end{aligned}$$

$m = 2$:

$$\begin{aligned} F(2) &= (1)\cdot(e^0) + (1)\cdot(e^{(-j2\pi/4)\times 2}) + (-1)\cdot(e^{(-j2\pi/4)\times 4}) + (-1)\cdot(e^{(-j2\pi/4)\times 6}) \\ &= 1 + \cos\pi - j\sin\pi - \cos 2\pi + j\sin 2\pi - \cos 3\pi + j\sin 3\pi \\ &= 1 - 1 - 0 - 1 + 0 + 1 + 0 = 0 \end{aligned}$$

$m = 3$:

$$\begin{aligned} F(3) &= (1)\cdot(e^0) + (1)\cdot(e^{(-j2\pi/4)\times 3}) + (-1)\cdot(e^{(-j2\pi/4)\times 6}) + (-1)\cdot(e^{(-j2\pi/4)\times 9}) \\ &= 1 + \cos\frac{3\pi}{2} - j\sin\frac{3\pi}{2} - \cos 3\pi + j\sin 3\pi - \cos\frac{3\pi}{2} + j\sin\frac{3\pi}{2} \\ &= 1 + 0 + j + 1 + 0 - 0 + j = 2 + j2 \end{aligned}$$

Check with MATLAB:

```
fn=[1  1  –1  –1]; Fm=fft(fn)

Fm =
  0    2.0000 - 2.0000i    0    2.0000 + 2.0000i
```

2.

$$x[0] = x[1] = x[2] = x[3] = 1 \text{ and } x[4] = x[5] = x[6] = x[7] = -1$$

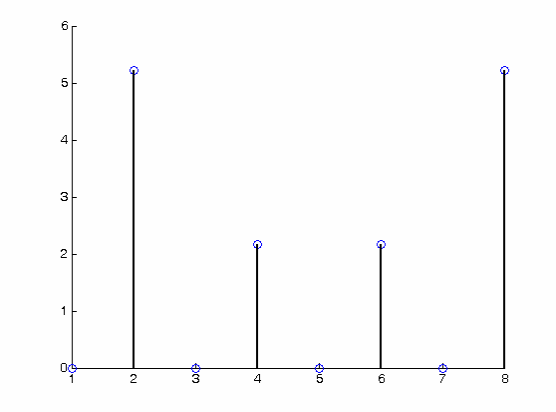
```
fn=[1 1 1 1 -1 -1 -1 -1]; magXm=abs(fft(fn)); fprintf(' \n');...
fprintf('magXm0 = %4.2f \t', magXm(1)); fprintf('magXm1 = %4.2f \t', magXm(2));...
fprintf('magXm2 = %4.2f \t', magXm(3)); fprintf('magXm3 = %4.2f \t', magXm(4)); fprintf('\n');...
fprintf('magXm4 = %4.2f \t', magXm(5)); fprintf('magXm5 = %4.2f \t', magXm(6));...
fprintf('magXm6 = %4.2f \t', magXm(7)); fprintf('magXm7 = %4.2f \t', magXm(8))
```

```
magXm0 = 0.00 magXm1 = 5.23 magXm2 = 0.00 magXm3 = 2.16
magXm4 = 0.00 magXm5 = 2.16 magXm6 = 0.00 magXm7 = 5.23
```

The MATLAB **stem** command can be used to plot discrete sequence data. For this Exercise we use the script

```
fn=[1 1 1 1 –1 –1 –1 –1]; stem(abs(fft(fn)))
```

and we obtain the plot below.



3.

$$x[n]\cos\frac{2\pi kn}{N} \Leftrightarrow \frac{1}{2}\{X[m-k]+X[m+k]\} \qquad x[n]\sin\frac{2\pi kn}{N} \Leftrightarrow \frac{1}{j2}\{X[m-k]+X[m+k]\}$$

From the frequency shift property of the DFT

$$W_N^{-km}x[n] \Leftrightarrow X[m-k] \quad (1)$$

Then,

$$W_N^{km}x[n] \Leftrightarrow X[m+k] \quad (2)$$

Adding (1) and (2) and multiplying the sum by $1/2$ we obtain

$$\frac{(W_N^{-km} + W_N^{km})(x[n])}{2} = \frac{(e^{j2\pi kn/N} + e^{-j2\pi kn/N})}{2}x[n] = x[n]\cos\frac{2\pi kn}{N}$$

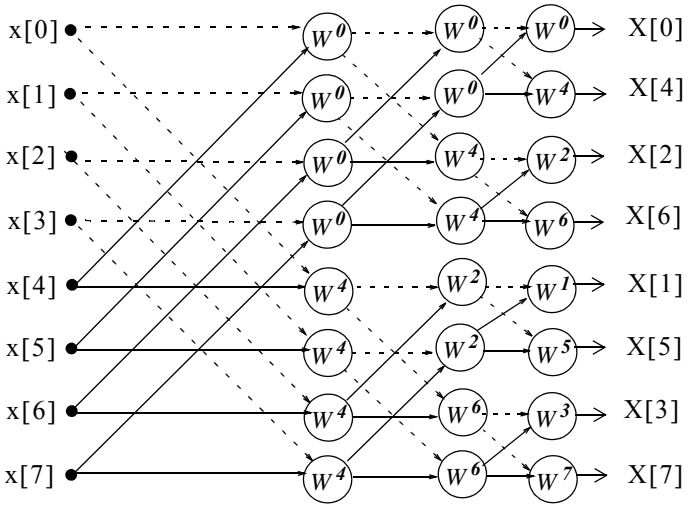
and thus

$$x[n]\cos\frac{2\pi kn}{N} \Leftrightarrow \frac{1}{2}[X[m-k] + X[m+k]]$$

Likewise, subtracting (2) from (1) and multiplying the difference by $1/j2$ we obtain

$$\frac{(W_N^{-km} - W_N^{km})(x[n])}{j2} = \frac{(e^{j2\pi kn/N} - e^{-j2\pi kn/N})}{j2}x[n] = x[n]\sin\frac{2\pi kn}{N}$$

4.



$$F(3) = X(3) = Y(6,3) = Y(6,2) + W_N^3 Y(7,2) \quad (1)$$

where

$$Y(6,2) = Y(4,1) + W_N^6 Y(6,1)$$

and

$$Y(7,2) = Y(5,1) + W_N^6 Y(7,1)$$

Going backwards (to the left) we find that

$$Y(4,1) = Y(0,0) + W_N^4 Y(4,0)$$

$$Y(6,1) = Y(2,0) + W_N^4 Y(6,0)$$

$$Y(5,1) = Y(1,0) + W_N^4 Y(5,0)$$

$$Y(7,1) = Y(3,0) + W_N^4 Y(7,0)$$

and by substitution into (1)

$$\begin{aligned} F(3) = X(3) &= Y(6,3) = Y(4,1) + W_N^6 Y(6,1) + W_N^3[Y(5,1) + W_N^6 Y(7,1)] \\ &= Y(0,0) + W_N^4 Y(4,0) + W_N^6[Y(2,0) + W_N^4 Y(6,0)] \\ &\quad + W_N^3\{Y(1,0) + W_N^4 Y(5,0) + W_N^6[Y(3,0) + W_N^4 Y(7,0)]\} \\ &= Y(0,0) + W_N^3 Y(1,0) + W_N^4 Y(4,0) + W_N^6 Y(2,0) \\ &\quad + W_N^7 Y(5,0) + W_N^9 Y(3,0) + W_N^{10} Y(6,0) + W_N^{13} Y(7,0) \end{aligned} \tag{2}$$

From the DFT definition

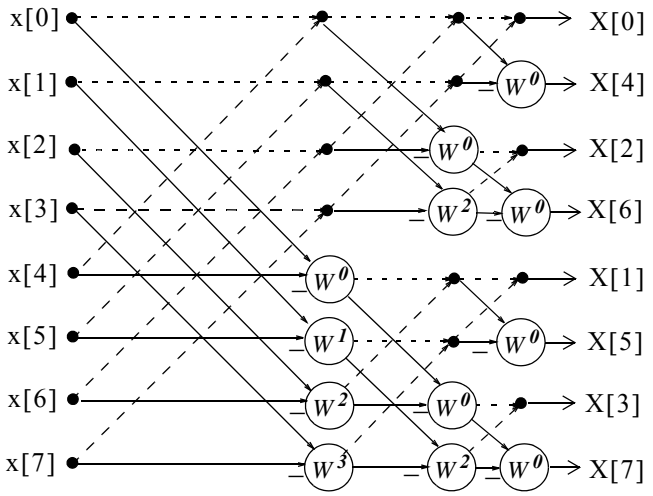
$$\begin{aligned} F(3) = X(3) &= x(0) + W_N^3 x(1) + W_N^6 x(2) + W_N^9 x(3) \\ &\quad + W_N^{12} x(4) + W_N^{15} x(5) + W_N^{18} x(6) + W_N^{21} x(7) \end{aligned} \tag{3}$$

By comparison, we see that the first 4 terms of (3) are the same the first, second, fourth, and sixth terms of (2) since $Y(k,0) = x(k)$, that is, $Y(0,0) = x(0)$, $Y(1,0) = x(1)$, and so on. The remaining terms in (2) and (3) are also the same since $W_8^{kN+r} = W_8^r$ and thus $W_8^{12} = W_8^4$, $W_8^{15} = W_8^7$, $W_8^{18} = W_8^{10}$, and $W_8^{21} = W_8^{13}$.

5.

$$Y_{dash}(R, C) = Y_{dash}(R_i, C-1) + Y_{dash}(R_j, C-1)$$

$$Y_{sol}(R, C) = W^m[Y_{sol}(R_i, C-1) - Y_{sol}(R_j, C-1)]$$



We are asked to compute F(3) only. However, we will derive all equations as we did in Example 10.5.

Column 1 (C=1):

$$\begin{aligned}
Y(0,1) &= Y(0,0) + Y(4,0)\\
Y(1,1) &= Y(1,0) + Y(5,0)\\
Y(2,1) &= Y(2,0) + Y(6,0)\\
Y(3,1) &= Y(3,0) + Y(7,0)\\
Y[4,1] &= W_N^0[Y(0,0) - Y(4,0)] \qquad (1)\\
Y[5,1] &= W_N^1[Y(1,0) - Y(5,0)]\\
Y[6,1] &= W_N^2[Y(2,0) - Y(6,0)]\\
Y[7,1] &= W_N^3[Y(3,0) - Y(7,0)]
\end{aligned}$$

Column 2 (C=2):

$$\begin{aligned}
Y(0,2) &= Y(0,1) + Y(2,1)\\
Y(1,2) &= Y(1,1) + Y(3,1)\\
Y(2,2) &= W_N^0[Y(0,1) - Y(2,1)]\\
Y(3,2) &= W_N^2[Y(1,1) - Y(3,1)] \qquad (2)\\
Y[4,2] &= Y(4,1) + Y(6,1)\\
Y[5,2] &= Y(5,1) + Y(7,1)\\
Y[6,2] &= W_N^0[Y(4,1) - Y(6,1)]\\
Y[7,2] &= W_N^2[Y(5,1) - Y(7,1)]
\end{aligned}$$

Column 3 (C=3):

$$\begin{aligned}
Y(0,3) &= Y(0,2) + Y(1,2)\\
Y(1,3) &= W_N^0[Y(0,2) - Y(1,2)]\\
Y(2,3) &= Y(2,2) + Y(3,2)\\
Y(3,3) &= W_N^0[Y(2,2) - Y(3,2)] \qquad (3)\\
Y[4,3] &= Y(4,2) + Y(5,2)\\
Y[5,3] &= W_N^0[Y(4,2) - Y(5,2)]\\
Y[6,3] &= Y(6,2) + Y(7,2)\\
Y[7,3] &= W_N^0[Y(6,2) - Y(7,2)]
\end{aligned}$$

$$F(3) = X(3) = Y(6,3) = Y(6,2) + Y(7,2) \qquad (4)$$

where

$$Y(6,2) = W_N^0[Y(4,1) - Y(6,1)]$$

and

$$Y(7,2) = W_N^2[Y(5,1) - Y(7,1)]$$

From (1)

$$Y[4,1] = W_N^0[Y(0,0) - Y(4,0)]$$

$$Y[5,1] = W_N^1[Y(1,0) - Y(5,0)]$$

$$Y[6,1] = W_N^2[Y(2,0) - Y(6,0)]$$

$$Y[7,1] = W_N^3[Y(3,0) - Y(7,0)]$$

and by substitution into (4)

$$\begin{aligned} F(3) = X(3) = {} & W_N^0 Y(0,0) + W_N^3 Y(1,0) - W_N^2 Y(2,0) - W_N^5 Y(3,0) \\ & - W_N^0 Y(4,0) - W_N^3 Y(5,0) + W_N^2 Y(6,0) + W_N^5 Y(7,0) \end{aligned} \tag{5}$$

From Exercise 4,

$$\begin{aligned} F(3) = X(3) = {} & x(0) + W_N^3 x(1) + W_N^6 x(2) + W_N^9 x(3) \\ & + W_N^{12} x(4) + W_N^{15} x(5) + W_N^{18} x(6) + W_N^{21} x(7) \end{aligned} \tag{6}$$

Since $Y(k,0) = x(k)$, $W_8^{8i+n} = W_8^n$ and $W_8^{n \pm 4} = -W_8^n$ (see proof below), we see that $W_8^6 = -W_8^2$, $W_8^9 = -W_8^5$, $W_8^{12} = -W_8^0$, $W_8^{15} = -W_8^3$, $W_8^{18} = W_8^2$, and $W_8^{21} = W_N^5$. Therefore, (5) and (6) are the same.

Proof that $W_8^{n \pm 4} = -W_8^n$:

$$W_8^{n \pm 4} = W_8^n \cdot W_8^{\pm 4} = W_8^n \cdot e^{-j2\pi/8 \cdot (\pm 4)} = W_8^n \cdot (\cos\pi \mp \sin\pi) = -W_8^n$$

6.

The rectangular pulse is produced with the MATLAB script below.

```
x=[linspace(–2,–1,100) linspace(–1,1,100) linspace(1,2,100)];...
y=[linspace(0,0,100) linspace(1,1,100) linspace(0,0,100)]; plot(x,y)
```

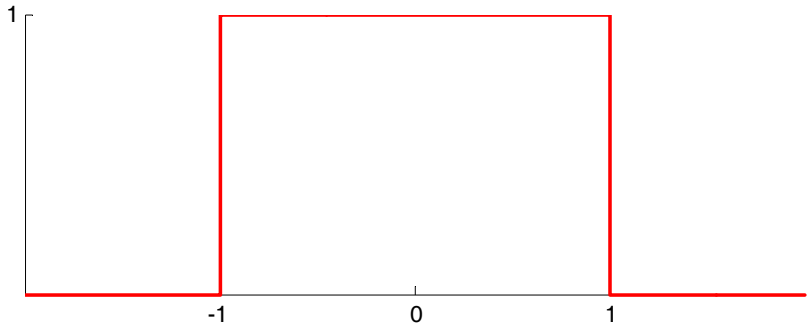
and the FFT is produced with

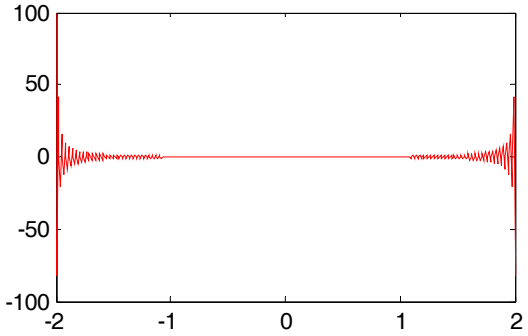
```
plot(x, fft(y))
```

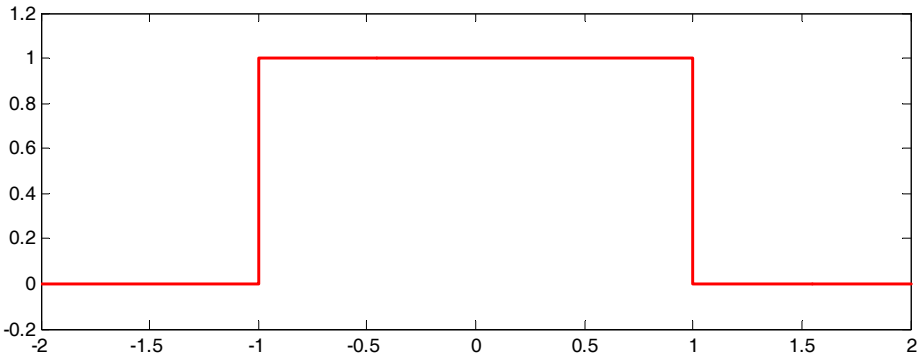
The Inverse FFT is produced with

```
plot(x,ifft(fft(y)))
```

The original rectangular pulse, its FFT, and the Inverse FFT are shown below.







7.

The triangular pulse is produced with the MATLAB script below.

```
x=linspace(–1,1,100); y=[linspace(0,1,50) linspace(1,0,50)]; plot(x,y)
```

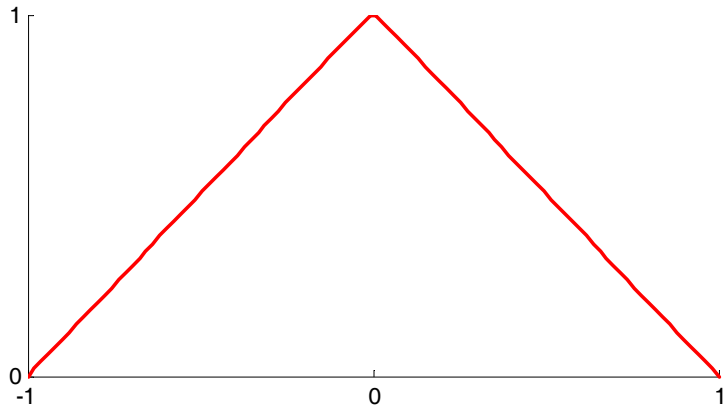
and the FFT is produced with

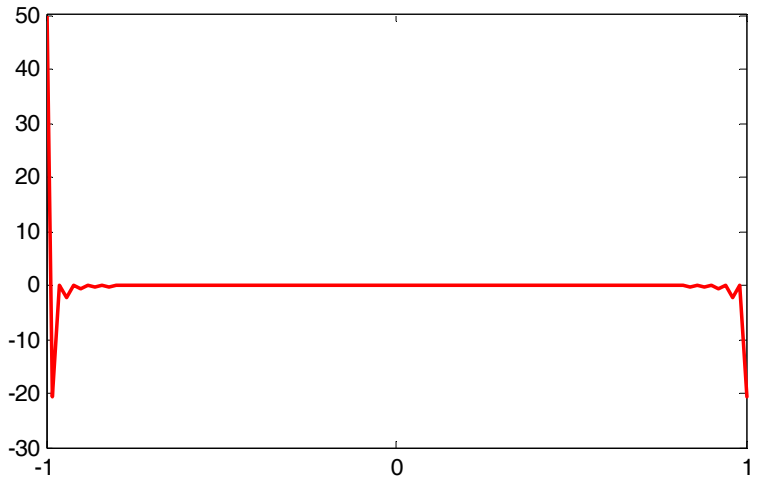
```
plot(x, fft(y))
```

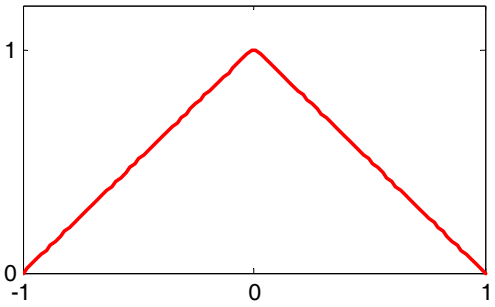
The Inverse FFT is produced with

```
plot(x,ifft(fft(y)))
```

The original triangular pulse, its FFT, and the Inverse FFT are shown below.







Chapter 11

Analog and Digital Filters

This chapter is an introduction to analog and digital filters. It begins with the basic analog filters, transfer functions, and frequency response. The magnitude characteristics of Butterworth and Chebychev filters and conversion of analog to equivalent digital filters using the bilinear transformation is presented. It concludes with design examples using MATLAB® and Simulink® when applicable.

11.1 Filter Types and Classifications

Analog filters are defined over a continuous range of frequencies. They are classified as *low–pass, high–pass, band–pass* and *band–elimination (stop–band)*. The ideal magnitude characteristics of each are shown in Figure 11.1. The ideal characteristics are not physically realizable; we will see that practical filters can be designed to approximate these characteristics.

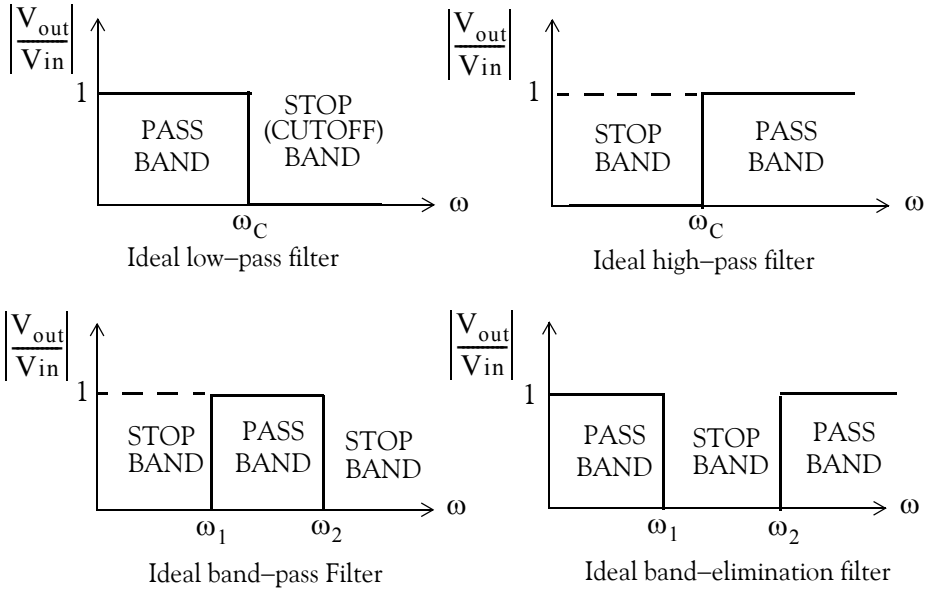


Figure 11.1. Magnitude characteristics of the types of filters

Another, less frequently mentioned filter, is the *all–pass* or *phase shift* filter. It has a constant magnitude response but is phase varies with frequency. Please refer to Exercise 4, Page 11–94, at the end of this chapter.

A *digital filter*, in general, is a computational process, or algorithm that converts one sequence of numbers representing the input signal into another sequence representing the output signal.

Accordingly, a digital filter can perform functions as differentiation, integration, estimation, and, of course, like an analog filter, it can filter out unwanted bands of frequency.

Analog filter functions have been used extensively as prototype models for designing digital filters and, therefore, we will present them first.

11.2 Basic Analog Filters

An analog filter can also be classified as *passive* or *active*. Passive filters consist of passive devices such as resistors, capacitors and inductors. Active filters are, generally, operational amplifiers with resistors and capacitors connected to them externally. We can find out whether a filter, passive or active, is a low–pass, high–pass, etc., from its the frequency response that can be obtained from its transfer function. The procedure is illustrated in Subsections 11.2.1 through 11.2.4 below.

11.2.1 RC Low–Pass Filter

The RC network shown in Figure 11.2 is a basic analog low–pass filter. We will derive expressions for its magnitude and phase.

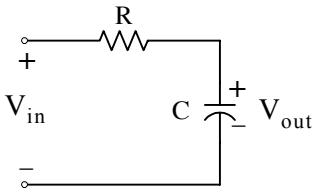


Figure 11.2. Basic low–pass RC network

Application of the voltage division expression yields

$$V_{out} = \frac{1/j\omega C}{R + 1/j\omega C} V_{in}$$

or

$$G(j\omega) = \frac{V_{out}}{V_{in}} = \frac{1}{1 + j\omega RC} = \frac{1}{(\sqrt{1 + \omega^2 R^2 C^2}) \angle \tan^{-1}(\omega RC)} = \frac{1}{\sqrt{1 + \omega^2 R^2 C^2}} \angle -\tan^{-1}(\omega RC) \quad (11.1)$$

The magnitude of (11.1) is

$$\boxed{|G(j\omega)| = \left|\frac{V_{out}}{V_{in}}\right| = \frac{1}{\sqrt{1 + \omega^2 R^2 C^2}}} \quad (11.2)$$

and the phase angle, also known as the *argument*, is

$$\boxed{\theta = \arg\{G(j\omega)\} = \arg\left(\frac{V_{out}}{V_{in}}\right) = -\tan^{-1}(\omega RC)} \quad (11.3)$$

We can obtain a quick sketch for the magnitude $|G(j\omega)|$ versus ω by evaluating (11.2) at $\omega = 0$, $\omega = 1/RC$, and $\omega \to \infty$. Thus,

as $\omega \to 0$,

$$|G(j\omega)| \cong 1$$

for $\omega = 1/RC$,

$$|G(j\omega)| = 1/\sqrt{2} = 0.707$$

and as $\omega \to \infty$,

$$|G(j\omega)| \cong 0$$

We will use the MATLAB script below to plot $|G(j\omega)|$ versus radian frequency ω. This is shown in Figure 11.3 where, for convenience, we let $RC = 1$.

```
w=0:0.02:10; RC=1; magGjw=1./sqrt(1+w.*RC); semilogx(w,magGjw);...
xlabel('Frequency in rad/sec – log scale'); ylabel('Magnitude of Vout/Vin');...
title('Magnitude Characteristics of basic RC low–pass filter'); grid
```

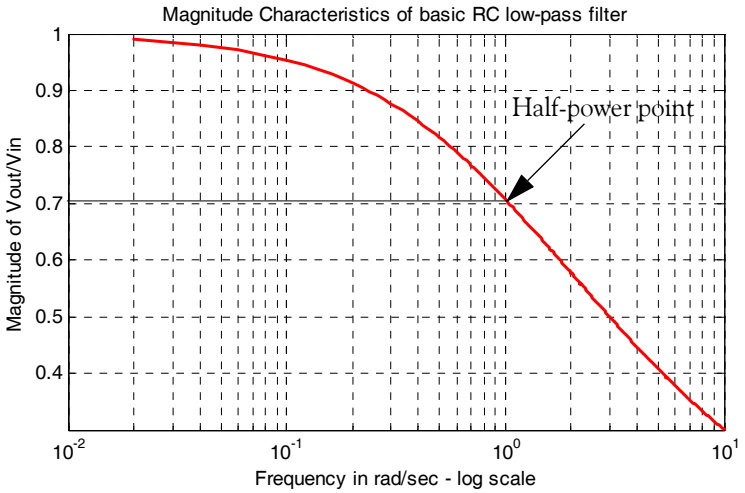


Figure 11.3. Magnitude characteristics of the basic RC low–pass filter with $RC = 1$

We can also obtain a quick sketch for the phase angle, i.e., $\theta = \arg\{G(j\omega)\}$ versus ω by evaluating (11.3) at $\omega = 0$, $\omega = 1/RC$, $\omega = -1/RC$, $\omega \to -\infty$ and $\omega \to \infty$. Thus,

As $\omega \to 0$,

$$\theta \cong -\text{atan}\,0 \cong 0°$$

For $\omega = 1/RC$,

$$\theta = -\text{atan}\,1 = -45°$$

For $\omega = -1/RC$,

$$\theta = -\text{atan}(-1) = 45°$$

As $\omega \rightarrow -\infty$,

$$\theta = -\text{atan}(-\infty) = 90°$$

and as $\omega \rightarrow \infty$,

$$\theta = -\text{atan}(\infty) = -90°$$

We will use the MATLAB script below to plot the phase angle θ versus radian frequency ω. This is shown in Figure 11.4 where, for convenience, we let $RC = 1$.

```
w=0:0.02:10; RC=1; phaseGjw=–atan(w.*RC).*180./pi; semilogx(w,phaseGjw);...
xlabel('Frequency in rad/sec – log scale'); ylabel('Phase of Vout/Vin – degrees');...
title('Phase Characteristics of basic RC low–pass filter'); grid
```

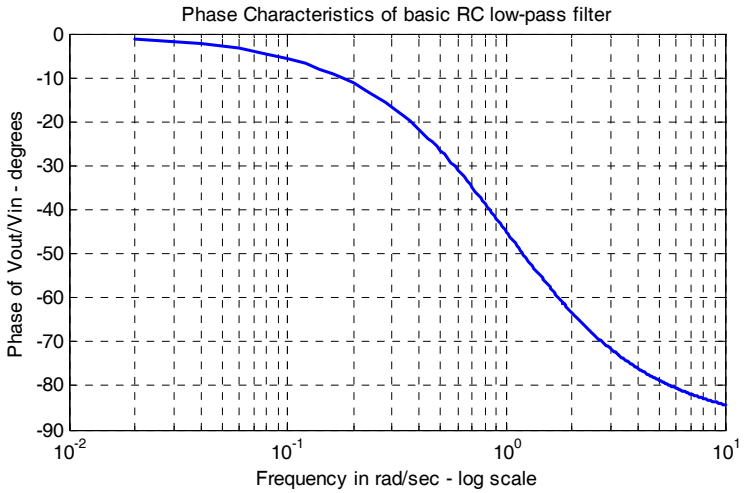


Figure 11.4. Phase characteristics of the basic RC low–pass filter with $RC = 1$

11.2.2 RC High–Pass Filter

The RC network shown in Figure 11.5 is a basic analog high–pass filter. We will derive expressions for its magnitude and phase.

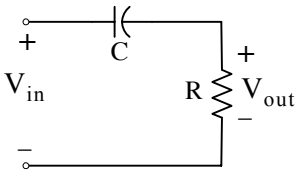


Figure 11.5. Basic high–pass RC network

Application of the voltage division expression yields

$$V_{out} = \frac{R}{R + 1/j\omega C} V_{in}$$

or

$$G(j\omega) = \frac{V_{out}}{V_{in}} = \frac{j\omega RC}{1+j\omega RC} = \frac{j\omega RC + \omega^2R^2C^2}{1+\omega^2R^2C^2} = \frac{\omega RC(j+\omega RC)}{1+\omega^2R^2C^2}$$
$$= \frac{\omega RC\sqrt{1+\omega^2R^2C^2}\angle\tan^{-1}(1/(\omega RC))}{1+\omega^2R^2C^2} = \frac{1}{\sqrt{1+1/(\omega^2R^2C^2)}}\angle\tan^{-1}(1/(\omega RC)) \tag{11.4}$$

The magnitude of (11.4) is

$$\boxed{|G(j\omega)| = \frac{1}{\sqrt{1+1/(\omega^2R^2C^2)}}} \tag{11.5}$$

and the phase angle or argument, is

$$\boxed{\theta = \arg\{G(j\omega)\} = \tan^{-1}(1/(\omega RC))} \tag{11.6}$$

We can obtain a quick sketch for the magnitude $|G(j\omega)|$ versus ω by evaluating (11.5) at $\omega = 0$, $\omega = 1/RC$, and $\omega \to \infty$. Thus,

As $\omega \to 0$,

$$|G(j\omega)| \cong 0$$

For $\omega = 1/RC$,

$$|G(j\omega)| = 1/\sqrt{2} = 0.707$$

and as $\omega \to \infty$,

$$|G(j\omega)| \cong 1$$

We will use the MATLAB script below to plot $|G(j\omega)|$ versus radian frequency ω. This is shown in Figure 11.6 where, for convenience, we let $RC = 1$.

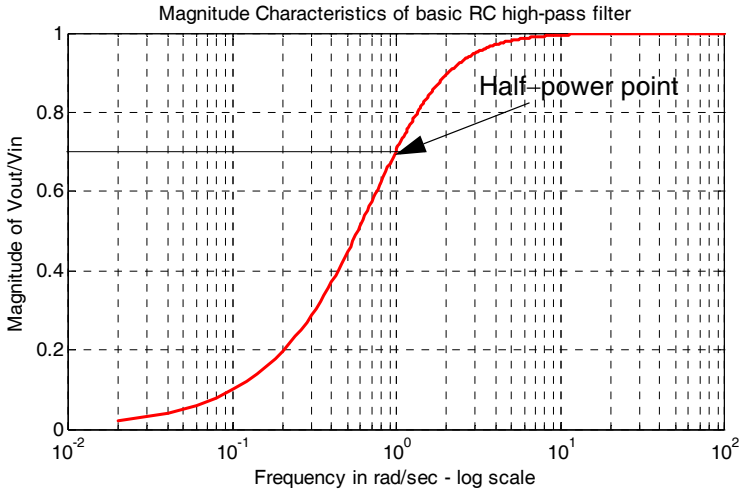


Figure 11.6. Magnitude characteristics of the basic RC high–pass filter with $RC = 1$

```
w=0:0.02:100; RC=1; magGjw=1./sqrt(1+1./(w.*RC).^2); semilogx(w,magGjw);...
xlabel('Frequency in rad/sec – log scale'); ylabel('Magnitude of Vout/Vin');...
title('Magnitude Characteristics of basic RC high–pass filter'); grid
```

We can also obtain a quick sketch for the phase angle, i.e., $\theta = \arg\{G(j\omega)\}$ versus ω, by evaluating (11.6) at $\omega = 0$, $\omega = 1/RC$, $\omega = -1/RC$, $\omega \to -\infty$, and $\omega \to \infty$. Thus,

As $\omega \to 0$,

$$\theta \cong -\text{atan}\,0 \cong 0°$$

For $\omega = 1/RC$,

$$\theta = -\text{atan}\,1 = -45°$$

For $\omega = -1/RC$,

$$\theta = -\text{atan}(-1) = 45°$$

As $\omega \to -\infty$,

$$\theta = -\text{atan}(-\infty) = 90°$$

and as $\omega \to \infty$,

$$\theta = -\text{atan}(\infty) = -90°$$

We will use the MATLAB script below to plot the phase angle θ versus radian frequency ω. This is shown in Figure 11.7 where, for convenience, we let $RC = 1$.

```
w=0:0.02:10; RC=1; phaseGjw=atan(1./(w.*RC)).*180./pi; semilogx(w,phaseGjw);...
xlabel('Frequency in rad/sec – log scale'); ylabel('Phase of Vout/Vin – degrees');...
title('Phase Characteristics of basic RC high–pass filter'); grid
```

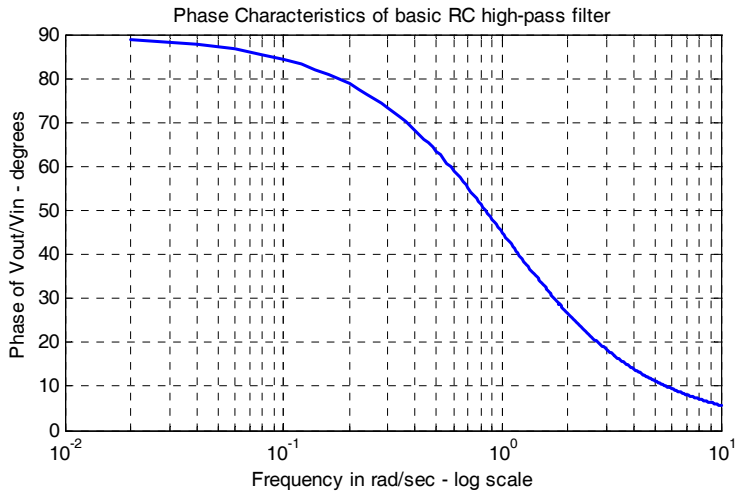


Figure 11.7. Phase characteristics of an RC high–pass filter with $RC = 1$

11.2.3 RLC Band–Pass Filter

The RLC network shown in Figure 11.8 is a basic analog band–pass filter.* We will derive expressions for its magnitude and phase.

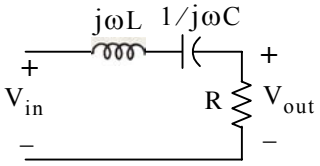


Figure 11.8. Basic band–pass RLC network

Application of the voltage division expression yields

$$V_{out} = \frac{R}{j\omega L + 1/j\omega C + R} \cdot V_{in}$$

and thus

$$G(j\omega) = \frac{V_{out}}{V_{in}} = \frac{R}{j\omega L + 1/j\omega C + R} = \frac{j\omega RC}{-\omega^2 LC + 1 + j\omega RC} = \frac{-j\omega(R/L)}{\omega^2 - j\omega(R/L) - 1/LC} \qquad (11.7)$$

There is no need to express relation (11.7) in magnitude and phase form. We will make use of the **abs(x)** and **angle(x)** MATLAB functions for the magnitude and phase plots.

We will use the MATLAB script below to plot $|G(j\omega)|$ versus radian frequency ω. This is shown in Figure 11.9 where, for convenience, we let $R = L = C = 1$, and thus (11.7) simplifies to:

$$G(j\omega) = \frac{-j\omega}{\omega^2 - j\omega - 1} \qquad (11.8)$$

```
w=0:0.02:100;  magGjw=abs(–j.*w./(w.^2–j.*w–1)); semilogx(w,magGjw);...
xlabel('Frequency in rad/sec – log scale'); ylabel('Magnitude of Vout/Vin');...
title('Magnitude Characteristics of basic RLC band–pass filter'); grid
```

The plot for the magnitude of (11.8) is shown in Figure 11.9.

To plot the phase of (11.8), we use the MATLAB script below.

```
w=0:0.02:100;  phaseGjw=angle(–j.*w./(w.^2–j.*w–1)).*180./pi; semilogx(w,phaseGjw);...
xlabel('Frequency in rad/sec – log scale'); ylabel('Phase of Vout/Vin – degrees');...
title('Phase Characteristics of basic RLC band–pass filter'); grid
```

The plot for the phase of (11.8) is shown in Figure 11.10.

* *This is the same network as (a) in Exercise 7, Chapter 4, Page 4–22.*

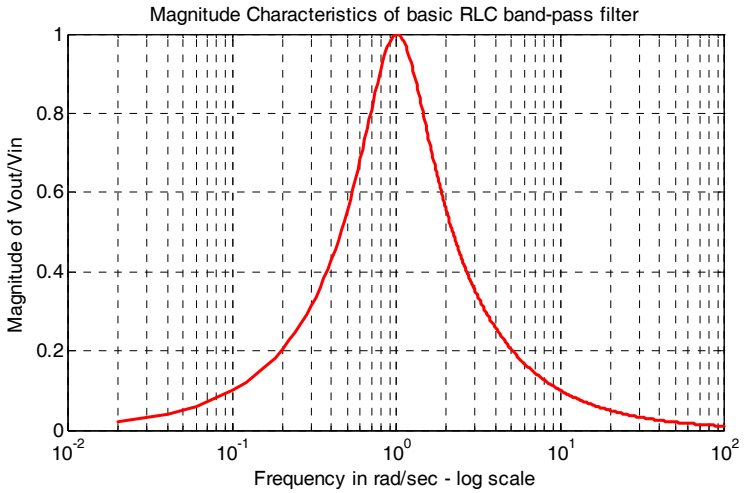


Figure 11.9. Magnitude characteristics of the basic RLC band–pass filter with R = L = C = 1

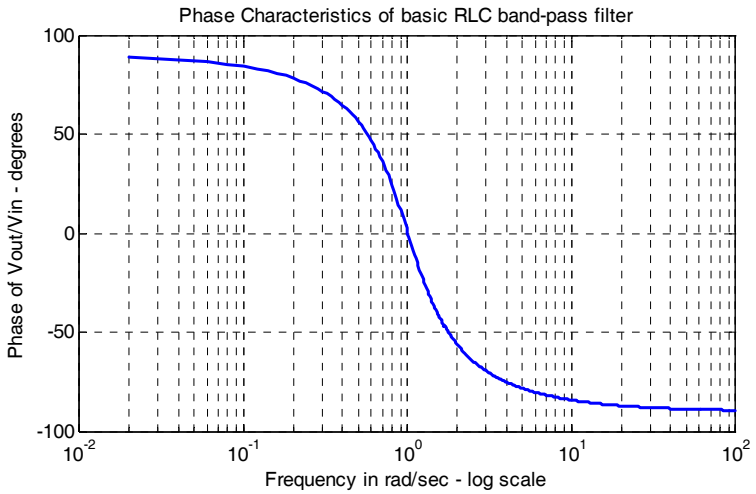


Figure 11.10. Phase characteristics of the basic RLC band–pass filter with R = L = C = 1

11.2.4 RLC Band–Elimination Filter

The RLC network shown in Figure 11.11 is a basic analog band–elimination filter.* We will derive expressions for its magnitude and phase.

Application of the voltage division expression yields

$$V_{out} = \frac{j\omega L + 1/j\omega C}{j\omega L + 1/j\omega C + R} \cdot V_{in}$$

* *This is the same network as (b) in Exercise 7, Chapter 4, Page 4–22. A band–elimination filter is also referred to as a band–stop filter or notch filter.*

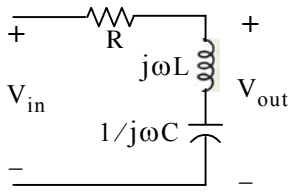


Figure 11.11. Basic band–elimination RLC network

and thus

$$G(j\omega) = \frac{V_{out}}{V_{in}} = \frac{j\omega L + 1/j\omega C}{j\omega L + 1/j\omega C + R} = \frac{-\omega^2 LC + 1}{-\omega^2 LC + 1 + j\omega RC} = \frac{\omega^2 - 1}{\omega^2 - j\omega(R/L) - 1/LC} \tag{11.9}$$

There is no need to express relation (11.9) in magnitude and phase form. As in the previous subsection, we will make use of the **abs(x)** and **angle(x)** MATLAB functions for the magnitude and phase plots.

We will use the MATLAB script below to plot $|G(j\omega)|$ versus radian frequency ω. For convenience, we let $R = L = C = 1$, and thus (11.9) simplifies to:

$$G(j\omega) = \frac{\omega^2 - 1}{\omega^2 - j\omega - 1} \tag{11.10}$$

```
w=0:0.02:100;  magG=abs((w.^2–1)./(w.^2–j.*w–1)); semilogx(w,magG); grid
```

The plot for the magnitude of (11.10) is shown in Figure 11.12.

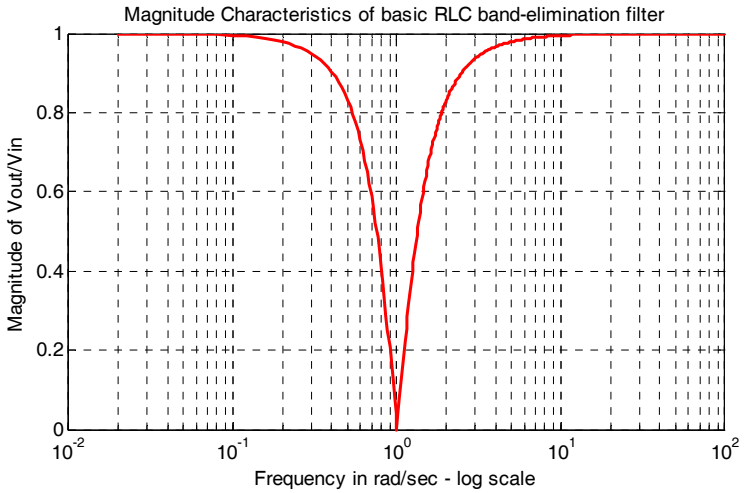


Figure 11.12. Magnitude characteristics of the basic RLC band–elimination filter with $R = L = C = 1$

To plot the phase of (11.10), we use the MATLAB script below.

```
w=0:0.02:100; phaseGjw=angle((w.^2–1)./(w.^2–j.*w–1)).*180./pi; semilogx(w,phaseGjw);...
```

```
xlabel('Frequency in rad/sec – log scale'); ylabel('Phase of Vout/Vin – degrees');...
title('Phase Characteristics of basic RLC band–elimination filter'); grid
```

The plot for the phase of (11.10) is shown in Figure 11.13.

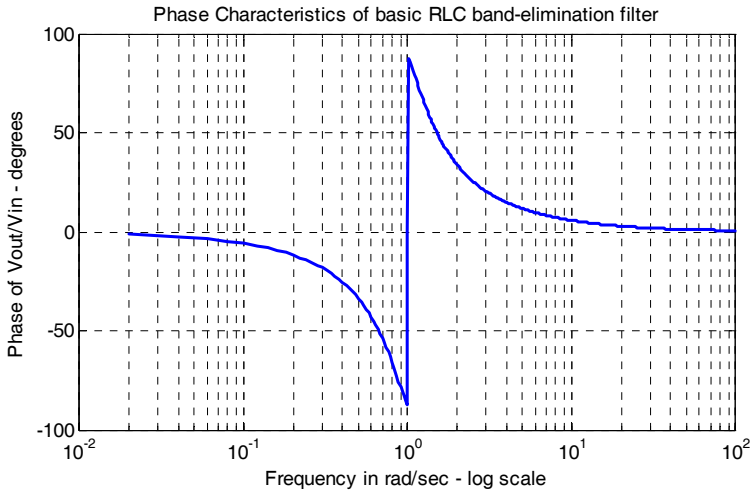


Figure 11.13. Phase characteristics of the basic RLC band–elimination filter with $R = L = C = 1$

Other passive filter networks are presented as end–of–chapter exercises in Chapter 4. Examples of active low–pass filters are presented in Figure 4.20, Chapter 4, Page 4–15, and at the end of this chapter. In the next section we introduce filter prototypes and use these as a basis for filter design.

11.3 Low–Pass Analog Filter Prototypes

In this section, we will use the analog low–pass filter as a basis. We will see later that, using transformations, we can derive high–pass and the other types of filters from a basic low–pass filter. We will discuss the *Butterworth, Chebyshev Type I, Chebyshev Type II also known as Inverted Chebyshev,* and *Cauer also known as elliptic* filters.

The first step in the design of an analog low–pass filter is to derive a suitable *magnitude–squared function* $A^2(\omega)$, and from it derive a $G(s)$ function such that

$$A^2(\omega) = G(s) \cdot G(-s)\big|_{s = j\omega} \tag{11.11}$$

Since $(j\omega)^* = (-j\omega)$, the square of the magnitude of a complex number can be expressed as that number and its complex conjugate. Thus, if the magnitude is A, then

$$A^2(\omega) = |G(j\omega)| = G(j\omega)G^*(j\omega) = G(j\omega) \cdot G(-j\omega) \tag{11.12}$$

Now, $G(j\omega)$ can be considered as $G(s)$ evaluated at $s = j\omega$, and thus (11.11) is justified. Also, since A is understood to represent the magnitude, it needs not be enclosed in vertical lines.

Not all magnitude–squared functions can be decomposed to $G(s)$ and $G(-s)$ rational functions; only even functions of ω, positive for all ω, and *proper rational functions** can satisfy (11.11).

Example 11.1

It is given that

$$G(s) = \frac{3s^2 + 5s + 7}{s^2 + 4s + 6}$$

Compute $A^2(\omega)$.

Solution:

Since

$$G(s) = \frac{3s^2 + 5s + 7}{s^2 + 4s + 6}$$

it follows that

$$G(-s) = \frac{3s^2 - 5s + 7}{s^2 - 4s + 6}$$

and

$$G(s) \cdot G(-s) = \frac{3s^2 + 5s + 7}{s^2 + 4s + 6} \cdot \frac{3s^2 - 5s + 7}{s^2 - 4s + 6} = \frac{9s^4 + 17s^2 + 49}{s^4 - 4s^2 + 36}$$

Therefore,

$$A^2(\omega) = G(s) \cdot G(-s)\big|_{s = j\omega} = \frac{9s^4 + 17s^2 + 49}{s^4 - 4s^2 + 36}\bigg|_{s = j\omega} = \frac{9\omega^4 - 17\omega^2 + 49}{\omega^4 + 4\omega^2 + 36}$$

The general form of the magnitude–square function $A^2(\omega)$ is

$$A^2(\omega) = \frac{C(b_k\omega^{2k} + b_{k-1}\omega^{2k-2} + \ldots + b_0)}{a_k\omega^{2k} + a_{k-1}\omega^{2k-2} + \ldots + a_0} \tag{11.13}$$

where C is the DC gain, a and b are constant coefficients, and k is a positive integer denoting the order of the filter. Once the magnitude–square function $A^2(\omega)$ is known, we can derive $G(s)$ from (11.11) with the substitution $(j\omega)^2 = -\omega^2 = s^2$ or $\omega^2 = -s^2$, that is,

* *It was stated earlier, that a rational function is said to be proper if the largest power in the denominator is equal to or larger than that of the numerator.*

$$G(s)\cdot G(-s) = A^2(\omega)\Big|_{\omega^2=-s^2} \tag{11.14}$$

In the simplest low–pass filter, the DC gain of the magnitude–square function is unity. In this case (11.13) reduces to

$$A^2(\omega) = \frac{b_0}{a_k\omega^{2k} + a_{k-1}\omega^{2k-2} + \ldots + a_0} \tag{11.15}$$

and at high frequencies can be approximated as

$$A^2(\omega) \approx \frac{b_0/a_k}{\omega^{2k}} \tag{11.16}$$

The *attenuation rate* of this approximation is $6k\ \text{dB/octave}$ or $20k\ \text{dB/decade}$. To understand this, let us review the definitions of *octave* and *decade.*

Consider two frequencies u_1 and u_2, and let

$$u_2 - u_1 = \log_{10}\omega_2 - \log_{10}\omega_1 = \log_{10}\frac{\omega_2}{\omega_1} \tag{11.17}$$

If these frequencies are such that $\omega_2 = 2\omega_1$, we say that these frequencies are separated by one octave, and if $\omega_2 = 10\omega_1$, we say that they are separated by one decade.

To compute the attenuation rate of (11.16), we take the square root of both sides. Then,

$$A(\omega) = \frac{\sqrt{b_0/a_k}}{\omega^k} = \frac{\text{Constant}}{\omega^k} = \frac{B}{\omega^k} \tag{11.18}$$

Taking the common log of both sides of (11.18) and multiplying by 20, we obtain

$$20\log_{10}A(\omega) = 20\log_{10}B - 20\log_{10}\omega^k = -20k\log_{10}\omega + 20\log_{10}B \tag{11.19}$$

or

$$A(\omega)_{\text{dB}} = -20k\log_{10}\omega + 20\log_{10}B \tag{11.20}$$

Relation (11.20) is is an equation of a straight line with $\text{slope} = -20k\ \text{dB/decade}$, and $\text{intercept} = B$ as shown in Figure 11.14.

The procedure of finding the transfer function $G(s)$ from the magnitude–square function $A^2(\omega)$, is illustrated with the Example 11.2 below.

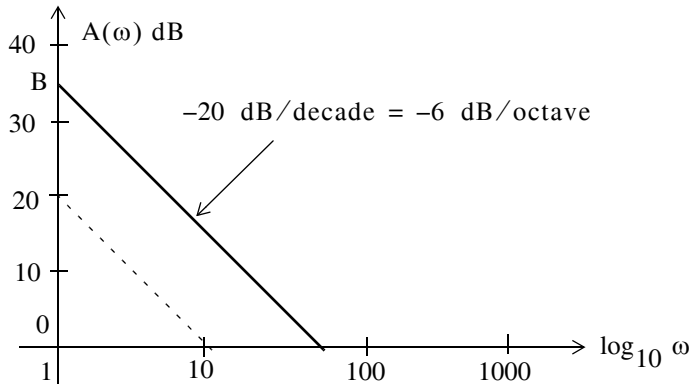


Figure 11.14. The –20 dB/decade = –6 dB/octave *line*

Example 11.2

Given the magnitude–square function

$$A^2(\omega) = \frac{16(-\omega^2 + 1)}{(\omega^2 + 4)(\omega^2 + 9)} \tag{11.21}$$

derive a suitable transfer function G(s)

Solution:

From relation (11.14),

$$G(s)G(-s) = A^2(\omega)\Big|_{\omega^2 = -s^2} = \frac{16(s^2 + 1)}{(-s^2 + 4)(-s^2 + 9)} \tag{11.22}$$

This function has zeros at $s = \pm j1$, and poles at $s = \pm 2$ and $s = \pm 3$.

There is no restriction on the zeros but, for stability*, we select the left–half s–plane poles. We must also select the gain constant such that $G(0) = A(0)$.

Let

$$G(s) = \frac{K(s^2 + 1)}{(s + 2)(s + 3)} \tag{11.23}$$

We must find K such that $G(0) = A(0)$. From (11.21),

* *Generally, a system is said to be stable if a finite input produces a finite output. Alternately, a system is stable if the impulse response* h(t) *vanishes after a sufficiently long time. Stability is discussed in Feedback and Control Systems textbooks.*

$$A^2(0) = 16/36 = 4/9$$

or

$$A(0) = 2/3$$

From (11.23),

$$G(0) = K/6$$

and for $G(0) = A(0)$ we must have,

$$K/6 = 2/3$$

or

$$K = 12/3 = 4$$

By substitution into (11.23), we obtain

$$G(s) = 4\frac{(s^2+1)}{(s+2)(s+3)}$$

11.3.1 Butterworth Analog Low-Pass Filter Design

In this subsection, we will consider the *Butterworth low–pass filter** whose magnitude–squared function is

$$A^2(\omega) = \frac{1}{(\omega/\omega_C)^{2k}+1} \tag{11.24}$$

where k is a positive integer, and ω_C is the cutoff (3 dB) frequency. Figure 11.15 is a plot of the relation (11.24) for $k = 1, 2, 4,$ and 8.

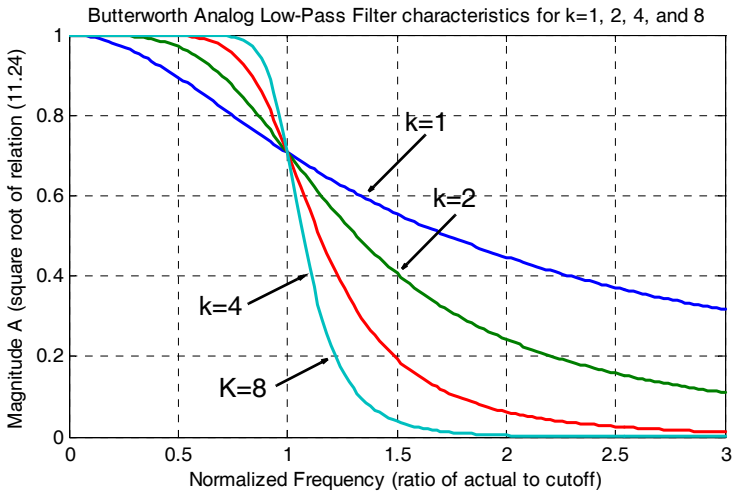


Figure 11.15. Butterworth low–pass filter magnitude characteristics

* *The frequency response of the Butterworth filter is maximally flat (has no ripples) in the passband, and rolls off towards zero in the stopband. When viewed on a logarithmic Bode plot, the response slopes off linearly towards negative infinity.*

The plot of Figure 11.15 was created with the following MATLAB script:

```
w_w0=0:0.02:3; Aw2k1=sqrt(1./(w_w0.^2+1)); Aw2k2=sqrt(1./(w_w0.^4+1));...
Aw2k4=sqrt(1./(w_w0.^8+1)); Aw2k8=sqrt(1./(w_w0.^16+1));...
plot(w_w0,Aw2k1,w_w0,Aw2k2,w_w0,Aw2k4,w_w0,Aw2k8);...
xlabel('Normalized Frequency (ratio of actual to cutoff)');...
ylabel('Magnitude A (square root of relation (11.24)');...
title('Butterworth Analog Low–Pass Filter characteristics for k=1, 2, 4, and 8'); grid
```

All Butterworth filters have the property that all poles of the transfer functions that describes them, lie on a circumference of a circle of radius ω_C, and they are $2\pi/2k$ radians apart. Thus, if $k = \text{odd}$, the poles start at zero radians, and if $k = \text{even}$, they start at $2\pi/2k$. But regardless whether k is odd or even, the poles are distributed in symmetry with respect to the $j\omega$ axis. For stability, we choose the left half–plane poles to form $G(s)$.

We can find the nth roots of a the complex number s by *DeMoivre's theorem*. This theorem states that

$$\sqrt[n]{re^{j\theta}} = \sqrt[n]{r}e^{j\left(\frac{\theta+2k\pi}{n}\right)} \qquad k = 0, \pm1, \pm2, \ldots \tag{11.25}$$

Example 11.3

Derive the transfer function $G(s)$ for the third order $(k = 3)$ Butterworth low–pass filter with *normalized cutoff frequency* $\omega_C = 1$ rad/s.

Solution:

With $k = 3$ and $\omega_C = 1$ rad/s, (11.24) simplifies to

$$A^2(\omega) = \frac{1}{\omega^6 + 1} \tag{11.26}$$

With the substitution $\omega^2 = -s^2$, (11.26) becomes

$$G(s) \cdot G(-s) = \frac{1}{-s^6 + 1} \tag{11.27}$$

Then, $s = \sqrt[6]{1}\angle 0°$ and by DeMoivre's theorem, with $n = 6$,

$$\sqrt[6]{1e^{j0}} = \sqrt[6]{1}e^{j\left(\frac{0+2k\pi}{6}\right)} \qquad k = 0, 1, 2, 3, 4, 5$$

Thus,

$$s_1 = 1\angle 0° = 1 \qquad s_2 = 1\angle 60° = \frac{1}{2}+j\frac{\sqrt{3}}{2} \qquad s_3 = 1\angle 120° = -\frac{1}{2}+j\frac{\sqrt{3}}{2}$$

$$s_4 = 1\angle 180° = -1 \qquad s_5 = 1\angle 240° = -\frac{1}{2}-j\frac{\sqrt{3}}{2} \qquad s_6 = 1\angle 300° = \frac{1}{2}-j\frac{\sqrt{3}}{2}$$

As expected, these six poles lie on the circumference of the circle with radius $\omega_C = 1$ as shown in Figure 11.16.

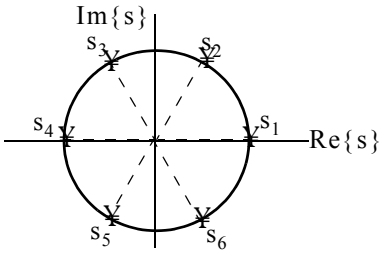


Figure 11.16. Location of the poles for the transfer function of Example 11.3

The transfer function $G(s)$ is formed with the left half–plane poles s_3, s_4, and s_5. Then,

$$G(s) = \frac{K}{\left(s+\frac{1}{2}-j\frac{\sqrt{3}}{2}\right)(s+1)\left(s+\frac{1}{2}+j\frac{\sqrt{3}}{2}\right)} \tag{11.28}$$

We use MATLAB to express the denominator as a polynomial.

```
syms s; den=(s+1/2–sqrt(3)*j/2)*(s+1)*(s+1/2+sqrt(3)*j/2)
```

```
den =
 (s+1/2-1/2*i*3^(1/2))*(s+1)*(s+1/2+1/2*i*3^(1/2))
```

```
expand(den)
```

```
ans =
s^3+2*s^2+2*s+1
```

Therefore, (11.28) simplifies to

$$G(s) = \frac{K}{s^3+2s^2+2s+1} \tag{11.29}$$

The gain K is found from $A^2(0) = 1$ or $A(0) = 1$ and $G(0) = K$. Thus, $K = 1$ and

$$G(s) = \frac{1}{s^3+2s^2+2s+1} \tag{11.30}$$

and this is the transfer function $G(s)$ for the third order $(k = 3)$ Butterworth low–pass filter with normalized cutoff frequency $\omega_C = 1$ rad/s.

The general form of any analog low–pass (Butterworth, Chebyshev, Elliptic, etc.) filter is

$$G(s)\big|_{lp} = \frac{b_0}{a_k s^k + \ldots + a_2 s^2 + a_1 s + a_0} \tag{11.31}$$

The pole locations and the coefficients of the corresponding denominator polynomials, have been derived and tabulated by Weinberg in *Network Analysis and Synthesis,* McGraw–Hill.

Table 11.1 shows the first through the fifth order coefficients for Butterworth analog low–pass filter denominator polynomials with normalized frequency $\omega_C = 1$ rad/s.

TABLE 11.1 Values for the coefficients a_i in (11.31)

Coefficients of Denominator Polynomial for Butterworth Low-Pass Filters						
Order	a_5	a_4	a_3	a_2	a_1	a_0
1						1
2				1	1.4142136	1
3			1	2	2	1
4		2.6131259	3.1442136	2.6131259	1	1
5	1	3.2360680	5.2360680	5.2360680	3.2360680	1

We can also use the MATLAB **buttap** and **zp2tf** functions to derive the coefficients. The **buttap** function returns the zeros, poles, and gain for an Nth order normalized prototype Butterworth analog low–pass filter. The resulting filter has N poles around the unit circle in the left half plane, and no zeros. The **zp2tf** function performs the zero–pole to transfer function conversion.

Example 11.4

Use MATLAB to derive the numerator b and denominator a coefficients for the third–order Butterworth low–pass filter prototype with normalized cutoff frequency*.

Solution:

```
[z,p,k]=buttap(3); [b,a]=zp2tf(z,p,k)

b =
     0     0     0     1
a =
    1.0000    2.0000    2.0000    1.0000
```

We observe that the denominator coefficients are the same as in Table 11.1.

* *Henceforth, normalized cutoff frequency will be understood to be* $\omega_C = 1$ rad/s

Table 11.2 lists the factored forms of the denominator polynomials in terms of linear and quadratic factors with normalized frequency $\omega_C = 1 \text{ rad/s}$.

TABLE 11.2 Factored forms for Butterworth low–pass filters

Denominator in Factored form for Butterworth Low–Pass Filters with $\omega_C = 1$ rad/s	
k	Denominator of Equation (11.27)
1	$s + 1$
2	$s^2 + 1.4142s + 1$
3	$(s + 1)(s^2 + s + 1)$
4	$(s^2 + 0.7654s + 1)(s^2 + 1.8478s + 1)$
5	$(s + 1)(s^2 + 0.6180s + 1)(s^2 + 1.6180s + 1)$
6	$(s^2 + 05176s + 1)(s^2 + 1.4142s + 1)(s^2 + 1.9318s + 1)$
7	$(s + 1)(s^2 + 0.4449s + 1)(s^2 + 1.2465s + 1)(s^2 + 1.8022s + 1)$
8	$(s^2 + 0.3896s + 1)(s^2 + 1.1110s + 1)(s^2 + 1.6630s + 1)(s^2 + 1.9622s + 1)$

The equations shown in Table 11.2 can be derived from

$$G(s) = \frac{1}{(-1)^n \prod_{i=0}^{n-1} \left(\frac{s}{s_i} - 1 \right)} \tag{11.32}$$

where the factor $(-1)^n$ is to ensure that $G(0) = 1$, and s_i denotes the poles on the left half of the s–plane. They can be found from

$$s_i = \omega_C \left(-\sin \frac{(2i + 1)\pi}{2k} + j \cos \frac{(2i + 1)\pi}{2k} \right) \tag{11.33}$$

We must remember that the factors in Table 11.2 apply only when the cutoff frequency is normalized to $\omega_C = 1 \text{ rad/s}$. If $\omega_C \neq 1$, we must scale the transfer function appropriately.

We can convert to the actual transfer function using the relation

$$G(s)_{\text{actual}} = G\left(\frac{\omega_{\text{norm}} \times s}{\omega_{\text{actual}}} \right)$$

and since, usually $\omega_{\text{norm}} = 1 \text{ rad/s}$,

$$G(s)_{actual} = G\left(\frac{s}{\omega_{actual}}\right) \tag{11.34}$$

that is, we replace s with s/ω_{actual}

Quite often, we require that $\omega \geq \omega_C$, that is, in the stop band of the low–pass filter, the attenuation to be larger than -20 dB/decade, i.e., we require a sharper cutoff. As we have seen from the plots of Figure 11.15, Page 11–14, the Butterworth low–pass filter cutoff becomes sharper for larger values of k. Accordingly, we generate the plot for different values of k shown in Figure 11.17 using the MATLAB script below.

```
w_w0=1:0.02:10; dBk1=20.*log10(sqrt(1./(w_w0.^2+1)));...
dBk2=20.*log10(sqrt(1./(w_w0.^4+1))); dBk3=20.*log10(sqrt(1./(w_w0.^6+1)));...
dBk4=20.*log10(sqrt(1./(w_w0.^8+1))); dBk5=20.*log10(sqrt(1./(w_w0.^10+1)));...
dBk6=20.*log10(sqrt(1./(w_w0.^12+1))); dBk7=20.*log10(sqrt(1./(w_w0.^14+1)));...
dBk8=20.*log10(sqrt(1./(w_w0.^16+1))); semilogx(w_w0,dBk1,w_w0,dBk2,w_w0,dBk3,...
w_w0,dBk4,w_w0,dBk5,w_w0,dBk6,w_w0,dBk7,w_w0,dBk8);...
xlabel('Normalized Frequency (rads/sec) – log scale'); ylabel ('Magnitude Response (dB)');...
title('Magnitude Attenuation as a Function of Normalized Frequency');...
set(gca, 'XTick', [1  2  3  4  5  6  7  8  9  10]); grid
```

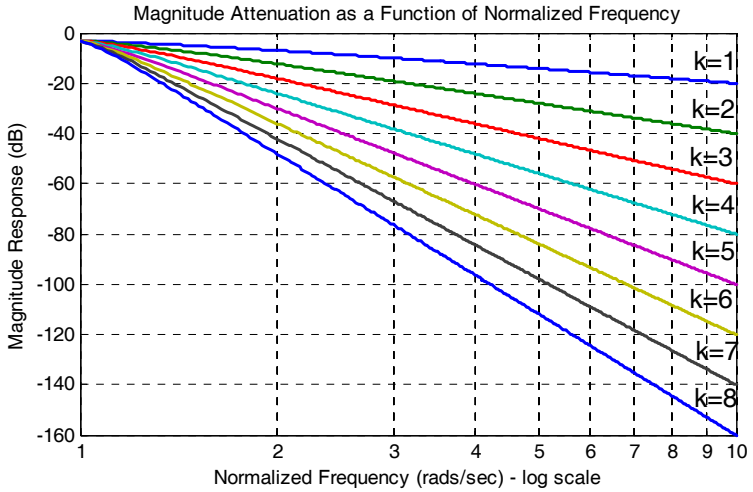


Figure 11.17. Attenuation for different values of k

Figure 11.17 indicates that for $k = 1$ the attenuation is -20 dB/decade, for $k = 2$ the attenuation is -40 dB/decade, and so on.

Example 11.5

Using the attenuation curves of Figure 11.17, derive the transfer function of a Butterworth low–pass analog filter with pass band bandwidth of $5\ \text{rad/s}$, and attenuation in the stop band at least $30\ \text{dB/decade}$ for frequencies larger than $15\ \text{rad/s}$.

Solution:

We refer to Figure 11.17 and at $\omega/\omega_C = 15/5 = 3$, we see that the vertical line at this value crosses the $k = 3$ curve at approximately $-28\ \text{dB}$, and the $k = 4$ curve at approximately $-37\ \text{dB}$. Since we require that the attenuation be at least $-30\ \text{dB}$, we use the attenuation corresponding to the $k = 4$ curve. Accordingly, we choose a fourth–order Butterworth low–pass filter whose normalized transfer function, from Table 11.2, is

$$G(s)_{norm} = \frac{1}{(s^2 + 0.7654s + 1)(s^2 + 1.8478s + 1)} \tag{11.35}$$

and since $\omega_C = 5\ \text{rad/s}$, we replace s with $s/5$. Then,

$$G(s)_{actual} = \frac{1}{\left(\frac{s^2}{25} + \frac{0.7654s}{5} + 1\right)\left(\frac{s^2}{25} + \frac{1.8478s}{5} + 1\right)}$$

$$G(s)_{actual} = \frac{625}{(s^2 + 3.8270s + 25)(s^2 + 9.2390s + 25)}$$

$$G(s)_{actual} = \frac{625}{s^4 + 13.066s^3 + 85.358s^2 + 326.650s + 625} \tag{11.36}$$

Of course, our objective is to learn how to design a circuit (passive or active), that will satisfy a transfer function such as the one above. Fortunately, the work for us has been done by others who have developed analog filter prototypes, both passive and active.

Some good references are:

Electronic Filter Design Handbook, Williams and Taylor, McGraw–Hill
Electronic Engineers' Handbook, Fink and Christiansen, McGraw–Hill
Reference Data for Engineers Handbook, Van Valkenburgh, Howard Sams

As an example, the *Reference Data for Engineers Handbook* provides the circuit of Figure 11.18 which is known as *Second Order Voltage Controlled Voltage Source (VCVS)* low–pass filter.

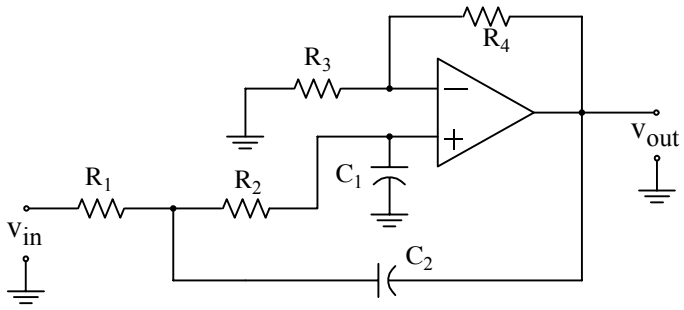


Figure 11.18. VCVS low–pass filter (Courtesy Reference Data for Engineers Handbook)

The transfer function of the second order VCVS low–pass filter of Figure 11.18 is given as

$$G(s) = \frac{Kb\omega_C^2}{s^2 + a\omega_C s + b\omega_C^2} \tag{11.37}$$

This is referred to as a second order *all–pole** approximation to the ideal low–pass filter with cut-off frequency ω_C, where K is the gain, and the coefficients a and b are provided by tables.

For a non-inverting positive gain K, the circuit of Figure 11.18 satisfies the transfer function of (11.37) with the conditions that

$$R_1 = \frac{2}{\left\{aC_2 + \left(\sqrt{[a^2 + 4b(K-1)]C_2^2 - 4bC_1C_2}\right)\right\}\omega_C} \tag{11.38}$$

$$R_2 = \frac{1}{bC_1C_2R_1\omega_C^2} \tag{11.39}$$

$$R_3 = \frac{K(R_1 + R_2)}{(K-1)} \qquad K \neq 1 \tag{11.40}$$

$$R_4 = K(R_1 + R_2) \tag{11.41}$$

From (11.40) and (11.41), we observe that $K = 1 + R_4/R_3$.

A fourth-order all–pole low–pass filter transfer function is a ratio of a constant to a fourth degree polynomial. A practical method of obtaining a fourth order transfer function, is to factor it into two second–order transfer functions of the form of relation (11.37), i.e.,

* *The terminology "all–pole" stems from the fact that the s–plane contains poles only and the zeros are at $\pm\infty$, that is, the s–plane is all poles and no zeros.*

$$G(s) = \frac{K_1 b_1 \omega_C^2}{s^2 + a_1 \omega_C s + b_1 \omega_C^2} \cdot \frac{K_2 b_2 \omega_C^2}{s^2 + a_2 \omega_C s + b_2 \omega_C^2} \tag{11.42}$$

Each factor in (11.42) can be realized by a stage (circuit). Then, the two stages can be cascaded as shown in Figure 11.19.

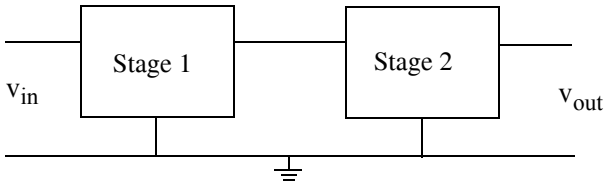


Figure 11.19. Cascaded stages

Table 11.3 lists the Butterworth low–pass coefficients for second and fourth–order designs, where a and b apply to the transfer functions of (11.37) and (11.42) respectively.

TABLE 11.3 Coefficients for Butterworth low–pass filter designs

Coefficients for Second and Fourth Order Butterworth Low–Pass Filter Designs		
Order		
2	a	1.41421
	b	1.0000
4	a_1	0.76537
	b_1	1.0000
	a_2	1.84776
	b_2	1.0000

For a practical design of a second–order VCVS circuit, we select standard values for capacitors C_1 and C_2 for the circuit of Figure 11.18, we substitute the appropriate values for the coefficients a and b from Table 11.3, we choose desired values for the gain K and cutoff frequency ω_C, and we substitute these in relations (11.38) through (11.41) to find the values of the resistors R_1 through R_4.

Example 11.6

Design a second–order VCVS Butterworth low–pass filter with gain $K = 2$ and cutoff frequency $f_C = 1$ KHz.

Solution:

We will use the second order VCVS prototype op amp circuit of Figure 11.18, with capacitance values $C_1 = C_2 = 0.01\ \mu F = 10^{-8}\ F$. From Table 11.3, $a = 1.41421 = \sqrt{2}$ and $b = 1$.

We substitute these values into (11.38) through (11.41), to find the values of the resistors.

We use MATLAB to do the calculations as follows:

```
C1=10^(–8); C2=C1; a=sqrt(2); b=1; K=2; wc=2*pi*10^3;...
% and from (11.34) through (11.37);...
R1=2/((a*C2+sqrt((a^2+4*b*(K–1))*C2^2–4*b*C1*C2))*wc);...
R2=1/(b*C1*C2*R1*wc^2);  R3=K*(R1+R2)/(K–1); R4=K*(R1+R2); fprintf(' \n');...
fprintf('R1 = %6.0f \t',R1); fprintf('R2 = %6.0f \t',R2);...
fprintf('R3 = %6.0f \t',R3); fprintf('R4 = %6.0f \t',R4)
```

```
R1 = 11254   R2 = 22508   R3 = 67524   R4 = 67524
```

These are the calculated values but they are not standard resistor values; we must select standard resistor values as close as possible to the calculated values.

It will be interesting to find out what the frequency response of this filter looks like, with capacitors $C_1 = C_2 = 0.01\ \mu F$ and standard 1% tolerance resistors with values $R_1 = 11.3\ K\Omega$, $R_2 = 2 \times R_1 = 22.6\ K\Omega$, and $R_3 = R_4 = 68.1\ K\Omega$.

We now substitute these values into the equations of (11.38) through (11.41), and we solve the first equation of this set for the cutoff frequency ω_C. Then, we use ω_C with the transfer function of (11.37). We do this with the following MATLAB script that produces the plot.

```
f=1:10:10^5; R1=11300; R2=22600; R3=68100; R4=R3; C1=10^(–8); C2=C1;...
a=sqrt(2); b=1; w=2*pi*f; fc=sqrt(1/(b*R1*R2*C1*C2))/(2*pi); wc=2*pi*fc;...
K=1+R3/R4; s=w*j; Gw=(K.*b.*wc.^2)./(s.^2+a.*wc.*s+b.*wc.^2);...
magGw=20.*log10(abs(Gw));...
semilogx(f,magGw); xlabel('Frequency Hz'); ylabel('|Vout/Vin| (dB)');...
title ('2nd Order Butterworth Low–Pass Filter Response'); grid
```

The frequency response of this low–pass filter is shown in Figure 11.20. We observe that the cutoff frequency occurs at about $1\ KHz$. As expected, the attenuation beyond $1\ KHz$ is at the rate of $–40\ dB/decade$ since this is a second–order low–pass filter. Also, since the circuit of a non–inverting op amp, its DC gain is 2 and thus $K_{dB} = 20\log 10(2) \approx 6$.

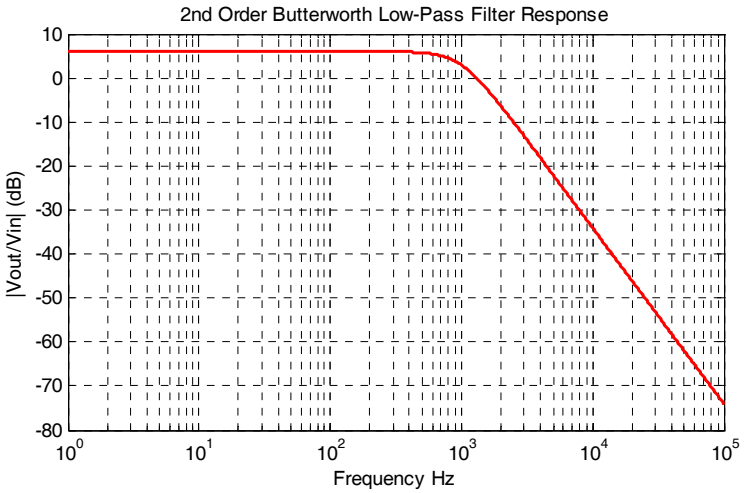


Figure 11.20. Plot for the VCVS low–pass filter of Example 11.6

We have used the MATLAB **buttap** function earlier to aid us in the design of Butterworth filters with the cutoff frequency normalized to 1 rad/s. We can also use the **bode** function to display both the (asymptotic) magnitude and phase plots. The following script will produce the Bode magnitude and phase plots for a two–pole Butterworth low-pass filter.

```
[z,p,k]= buttap(2);                    % Specify a two–pole filter;...
[b,a]=zp2tf(z,p,k);                    % Display in polynomial rational form;...
w=0:0.01:4; [mag,phase]=bode(b,a,w);...
b,a                                    % Display b and a coefficients
```

```
b =
     0     0     1

a =
    1.0000    1.4142    1.0000
```

```
num=[0  0  1]; den=[1  sqrt(2)  1];...
bode(num,den); title('Butterworth 2nd Order Low–Pass Filter'); grid
```

The Bode plots are shown in Figure 11.21. The frequency is displayed in rad/sec and the cutoff frequency normalized to 1 rad/s.

We can also display the Bode plots with the frequency specified in Hz. This can be done with the MATLAB script below.

```
h=bodeplot(tf(num,den));...
setoptions(h,'FreqUnits', 'Hz'); grid
```

The Bode plots with the frequency specified in Hz are shown in Figure 11.22.

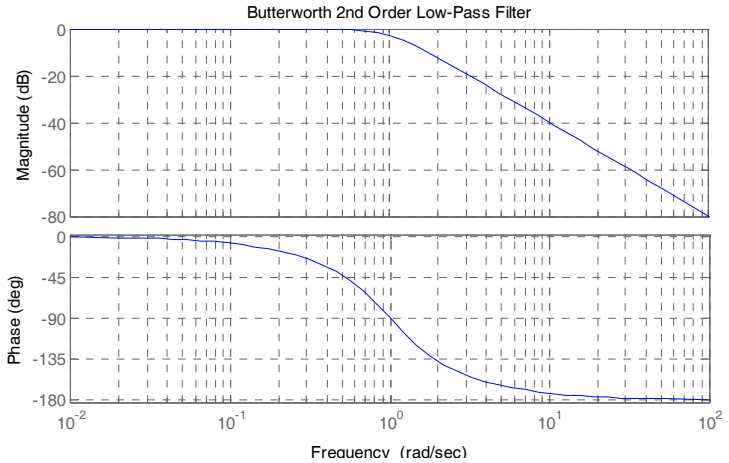


Figure 11.21. Bode plots for example 11.6 using MATLAB's bode function in rad/sec

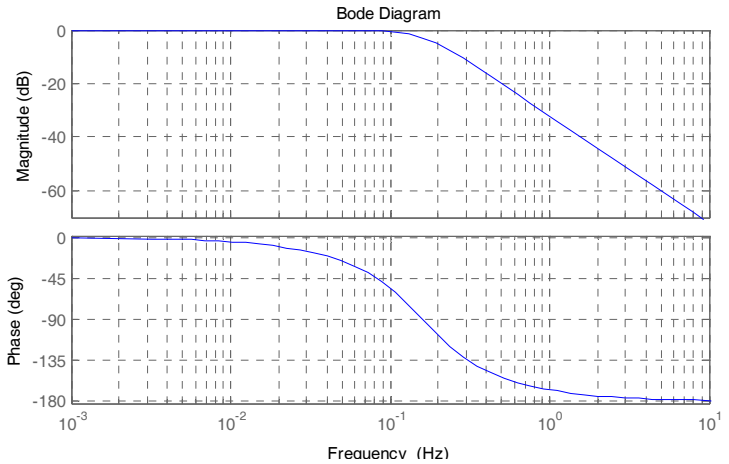


Figure 11.22. Bode plots for example 11.6 using MATLAB's bode function in Hz

11.3.2 Chebyshev Type I Analog Low–Pass Filter Design

The *Chebyshev Type I filters* are based on approximations derived from the Chebyshev polynomials $C_k(x)$ which constitute a set of orthogonal functions.* The coefficients of these polynomials are tabulated in math tables. See, for example, the *Handbook of Mathematical Functions*, Dover Publications. These polynomials are derived from the equations

* *Two functions are said to be orthogonal if, when multiplied together and integrated over the domain of interest, the integral becomes zero. The property of orthogonality is usually applied to a class of functions that differ by one or more variables.*

$$C_k(x) = \cos(k\cos^{-1}x) \quad (|x| \le 1) \tag{11.43}$$

and

$$C_k(x) = \cosh(k\cosh^{-1}x) \quad |x| > 1 \tag{11.44}$$

From (11.43), with $k = 0$, we obtain

$$C_0(x) = \cos(0\cos^{-1}x) = 1 \tag{11.45}$$

With $k = 1$,

$$C_1(x) = \cos(1\cos^{-1}x) = x^{*} \tag{11.46}$$

With $k = 2$

$$C_2(x) = \cos(2\cos^{-1}x) = 2x^2 - 1 \tag{11.47}$$

and this is shown by letting $\cos^{-1}x = \alpha$. Then,

$$C_2(x) = \cos(2\alpha) = 2\cos^2\alpha - 1 = 2\cos^2(\cos^{-1}x) - 1$$

$$= 2\left[\underbrace{\cos(\cos^{-1}x)}_{x}\underbrace{\cos(\cos^{-1}x)}_{x} - 1\right] = 2x^2 - 1$$

We can also use MATLAB to convert these trigonometric functions to algebraic polynomials. For example,

```
syms x; expand(cos(2*acos(x)))
```

```
ans =
2*x^2-1
```

Using this iterated procedure we can show that with $k = 3, 4,$ and 5, we obtain

$$C_3(x) = 4x^3 - 3x \qquad C_4(x) = 8x^4 - 8x^2 + 1 \qquad C_5(x) = 16x^5 - 20x^3 + 5x \tag{11.48}$$

and so on.

We observe that for $k = \text{even}$, $C_k(x) = \text{even}$, and for $k = \text{odd}$, $C_k(x) = \text{odd}$.

The curves representing these polynomials are shown in Figure 11.23.

The Chebyshev Type I low–pass filter magnitude–square function is defined as

$$A^2(\omega) = \frac{\alpha}{1 + \varepsilon^2 C_k^2(\omega/\omega_C)} \tag{11.49}$$

---

* *We* recall that if $x = \cos y$, then $y = \cos^{-1}x$, and $\cos y = x$.

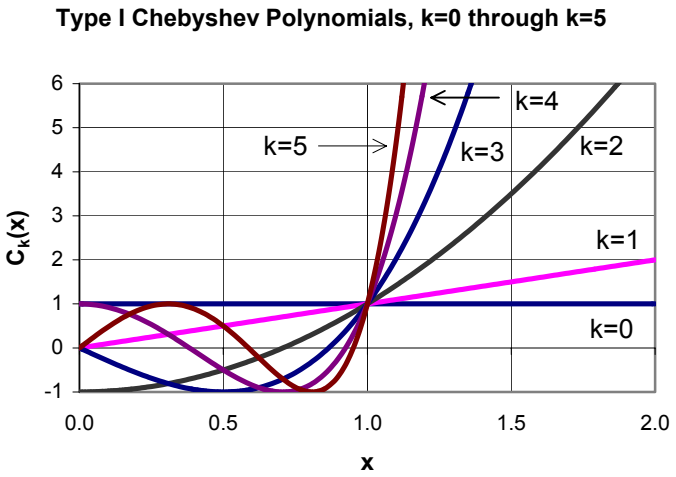


Figure 11.23. Chebyshev Type I polynomials

In relation (11.49), the quantity ε^2 is a parameter chosen to provide the desired pass–band ripple, the parameter α is a constant chosen to determine the desired DC gain, the subscript k denotes both the degree of the Chebyshev Type I polynomial and the order of the transfer function, and ω_C is the cutoff frequency. This filter produces a sharp cutoff rate in the transition band.

Figure 11.24 shows Chebyshev Type I magnitude frequency responses for $k = 3$ and $k = 4$.

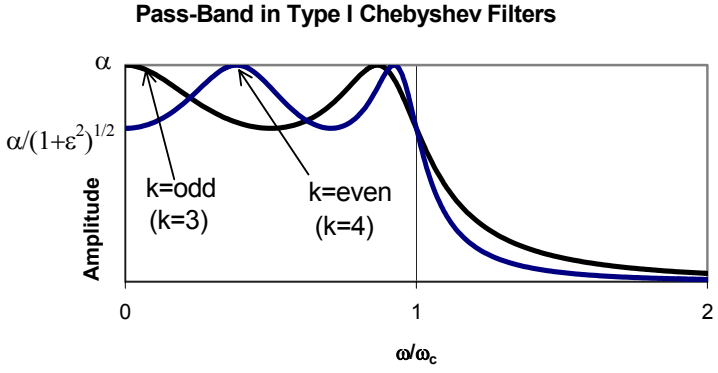


Figure 11.24. Chebyshev Chebyshev Type I low–pass filter for even and odd values of k.

The magnitude at $\omega = 0$ is α when $k = \text{odd}$ and it is $\alpha/\sqrt{1+\varepsilon^2}$ when $k = \text{even}$. This is

shown in Figure 11.24. The cutoff frequency is the largest value of ω_C for which

$$A(\omega_C) = \frac{1}{\sqrt{1+\varepsilon^2}} \tag{11.50}$$

Stated in other words, the pass–band is the range over which the ripple oscillates with constant bounds; this is the range from DC to ω_C. From (11.50), we observe that only when $\varepsilon = 1$ the magnitude at the cutoff frequency is 0.707 i.e., the same as in other types of filters. But when $0 < \varepsilon < 1$, the cutoff frequency is greater than the conventional 3 dB cutoff frequency ω_C.

Table 11.4 gives the ratio of the conventional cutoff frequency $f_{3\ dB}$ to the ripple width frequency f_C of a Chebyshev Type I low–pass filter.

TABLE 11.4 Ratio of conventional cutoff frequency to ripple width frequency

Ratio of Conventional $f_{3\ dB}$ Cutoff Frequency to Ripple Width for Low–Pass Chebyshev Filters		
Ripple Width	$f_{3\ dB}/f_c$	
dB	k=2	k=4
0.1	1.943	1.213
0.5	1.390	1.093
1.0	1.218	1.053

The pass–band ripple r in dB, is defined as

$$r_{dB} = 10\log_{10}\frac{A_{max}^2}{A_{min}^2} = 20\log_{10}\frac{A_{max}}{A_{min}} \tag{11.51}$$

where A_{max} and A_{min} are the maximum and minimum values respectively of the magnitude A in the pass–band interval. From (11.49),

$$A^2(\omega) = \frac{\alpha}{1+\varepsilon^2 C_k^2(\omega/\omega_C)} \tag{11.52}$$

and A_{max}^2 occurs when $\varepsilon^2 C_k^2(\omega/\omega_C) = 0$. Then,

$$A_{max}^2 = \alpha \tag{11.53}$$

To find A_{min}^2, we must first confirm that

$$C_k^2(\omega/\omega_C) \le 1$$

This can be shown to be true by (11.43), that is,

$$C_k(x) = \cos(k\cos^{-1}x) \quad |x| \le 1$$

or

$$|C_k(x)| \le 1 \quad \text{for} \quad -1 \le x \le 1$$

Therefore,

$$C_k^2(\omega/\omega_C)_{max} = 1$$

and

$$A_{min}^2 = \frac{\alpha}{1+\varepsilon^2} \tag{11.54}$$

Substitution of (11.53) and (11.54) into (11.51) yields

$$r_{dB} = 10\log_{10}\frac{A_{max}^2}{A_{min}^2} = 10\log_{10}\left[\frac{\alpha}{\alpha/(1+\varepsilon^2)}\right] = 10\log_{10}(1+\varepsilon^2) \tag{11.55}$$

or

$$\log_{10}(1+\varepsilon^2) = \frac{r_{dB}}{10}$$

$$1+\varepsilon^2 = 10^{r_{dB}/10}$$

or

$$\boxed{\varepsilon^2 = 10^{r_{dB}/10} - 1} \tag{11.56}$$

We have seen that when $k = \text{odd}$, there is a maximum at $\omega = 0$. At this frequency, (11.49) reduces to

$$A^2(0) = \alpha \tag{11.57}$$

and for a unity gain, $\alpha = 1$ when $k = \text{odd}$.

However, for unity gain when $k = \text{even}$, we must have $\alpha = 1+\varepsilon^2$. This is because at $\omega = 0$, we must have $C_k(0) = 1$ in accordance with(11.45). Then, the relation

$$A^2(\omega) = \frac{\alpha}{1+\varepsilon^2 C_k^2(\omega/\omega_C)}$$

reduces to

$$A^2(0) = \frac{\alpha}{1+\varepsilon^2 C_k^2(0)} = \frac{\alpha}{1+\varepsilon^2} = 1$$

or

$$\alpha = 1+\varepsilon^2$$

For this choice of α, the magnitude response at maxima, corresponds to

$$A^2(\omega_{max}) = \frac{1+\varepsilon^2}{1+\varepsilon^2 C_k^2(\omega_{max}/\omega_C)}$$

and this will be maximum when

$$C_k^2(\omega_{max}/\omega_C) = 0$$

resulting in

$$A^2(\omega_{max}) = \frac{1+\varepsilon^2}{1} = 1+\varepsilon^2$$

or

$$A(\omega_{max}) = \sqrt{1+\varepsilon^2}$$

Example 11.7

Derive the transfer function $G(s)$ for the $k = 2$, Chebyshev Type I function that has pass–band ripple $r_{dB} = 1$ dB, unity DC gain, and normalized cutoff frequency at $\omega_C = 1$ rad/s.

Solution:

From (11.49),

$$A^2(\omega) = \frac{\alpha}{1+\varepsilon^2 C_k^2(\omega/\omega_C)} \tag{11.58}$$

and since $k = \text{even}$, for unity DC gain, we must have $\alpha = 1+\varepsilon^2$. Then, (11.58) becomes

$$A^2(\omega) = \frac{1+\varepsilon^2}{1+\varepsilon^2 C_k^2(\omega/\omega_C)}$$

For $k = 2$

$$C_2(x) = 2x^2 - 1$$

and

$$C_k^2(\omega/\omega_C) = C_k^2(\omega) = (2\omega^2 - 1)^2 = 4\omega^4 - 4\omega + 1$$

Also, from (11.56),

$$\varepsilon^2 = 10^{r_{dB}/10} - 1 = 10^{1/10} - 1 = 1.259 - 1 = 0.259$$

Then,

$$A^2(\omega) = \frac{1+0.259}{1+0.259(4\omega^4 - 4\omega + 1)} = \frac{1.259}{1.036\omega^4 - 1.036\omega^2 + 1.259}$$

and with $\omega^2 = -s^2$,

$$G(s)G(-s) = \frac{1.259}{1.036s^4 + 1.036s^2 + 1.259}$$

We find the poles from the roots of the denominator using the MATLAB script below.

```
d=[1.036  0  1.036  0  1.259]; p=roots(d); fprintf(' \n'); disp('p1 = '); disp(p(1));...
disp('p2 = '); disp(p(2)); disp('p3 = '); disp(p(3)); disp('p4 = '); disp(p(4))
```

```
p1 =

  -0.5488 + 0.8951i

p2 =

  -0.5488 - 0.8951i

p3 =

   0.5488 + 0.8951i

p4 =

   0.5488 - 0.8951i
```

We now form the transfer function from the left half–plane poles $p_1 = -0.5488 + j0.8951$ and $p_2 = -0.5488 - j0.8951$. Then,

$$G(s) = \frac{K}{(s-p_1)(s-p_2)} = \frac{K}{(s+0.5488-j0.8951)(s+0.5488+j0.8951)}$$

We will use MATLAB script below to multiply the factors of the denominator.

```
syms s; den=(s+0.5488–0.8951*j)*(s+0.5488+0.8951*j); simple(expand(den))
```

```
ans =
s^2+686/625*s+22047709/20000000
```

```
686/625
```

```
ans =
    1.0976
```

```
22047709/20000000
```

```
ans =
    1.1024
```

Thus,

$$G(s) = \frac{K}{s^2 + 1.0976s + 1.1024}$$

and at $s = 0$,

$$G(0) = \frac{K}{1.1024}$$

Also, $A^2(0) = 1$, or $A(0) = 1$

Then,

$$G(0) = A(0) = \frac{K}{1.1024} = 1$$

or

$$K = 1.1024$$

Therefore, the transfer function for Example 11.7 is

$$G(s) = \frac{1.1024}{s^2 + 1.0976s + 1.1024}$$

We can plot the attenuation band for Chebyshev Type I filters, as we did with the Butterworth filters in Figure 11.17, but we need to construct one for each value of dB in the ripple region. Instead, we will develop the following procedure.

We begin with the Chebyshev approximation

$$A^2(\omega) = \frac{\alpha}{1 + \varepsilon^2 C_k^2(\omega/\omega_C)} \tag{11.59}$$

and, for convenience, we let $\alpha = 1$. If we want the magnitude of this to be less than some value β for $\omega \geq \omega_C$, we should choose the value of k in $C_k^2(\omega/\omega_C)$ so that

$$\frac{1}{1 + \varepsilon^2 C_k^2(\omega/\omega_C)} \leq \beta^2 \tag{11.60}$$

that is, we need to find a suitable value of the integer k so that (11.60) will be satisfied. As we have already seen from (11.56), the value of ε can be determined from

$$\varepsilon^2 = 10^{r_{dB}/10} - 1$$

once the band–pass ripple has been specified.

Next, we need to find G(s) from

$$A^2(\omega) = G(s)G(-s)\big|_{s = j\omega}$$

and if we replace ω by s/j in (11.59) where $\alpha = 1$, we obtain

$$|G(s)|^2 = \frac{1}{1 + \varepsilon^2 C_k^2(s/j\omega_C)} \tag{11.61}$$

It can be shown that the poles of the left–half of the s–plane are given by

$$s_i = \omega_C\left[-b\sin\frac{(2i+1)\pi}{2k} + jc\cos\frac{(2i+1)\pi}{2k}\right] \tag{11.62}$$

for $i = 0, 1, 2, \ldots, 2k-1$

The constants b and c in (11.62) can be evaluated from

$$b = \frac{m - m^{-1}}{2} \tag{11.63}$$

and

$$c = \frac{m + m^{-1}}{2} \tag{11.64}$$

where

$$m = \left(\sqrt{1 + \varepsilon^{-2}} + \varepsilon^{-1}\right)^{1/k} \tag{11.65}$$

The transfer function is then computed from

$$G(s) = \frac{(-1)^k}{\prod_{i=0}^{k-1}\left(\frac{s}{s_i} - 1\right)} \tag{11.66}$$

Example 11.8

Design a Chebyshev Type I analog low–pass filter with 3 dB band–pass ripple and $\omega_C = 5$ rad/s. The attenuation for $\omega \geq 15$ rad/s must be at least 30 dB/decade.

Solution:

From (11.56),

$$\varepsilon^2 = 10^{r_{dB}/10} - 1 = 10^{3/10} - 1 = 1.9953 - 1 \approx 1$$

and the integer k must be chosen such that

$$10\log_{10}\frac{1}{1 + C_k^2(15/5)} \leq -30$$

or

$$-10\log_{10}(1+C_k^2(3)) \le -30$$

$$-\log_{10}(1+C_k^2(3)) \le -3$$

$$1+C_k^2(3) \ge 10^3$$

To find the minimum value of k which satisfies this inequality, we compute the Chebyshev polynomials for $k = 0, 1, 2, 3, \ldots$. From (11.45) through (11.48), we obtain

$$C_0^2(3) = 1$$

$$C_1^2(3) = 3^2 = 9$$

$$C_2^2(3) = (2\cdot 3^2 - 1)^2 = 17^2 = 289$$

$$C_3^2(3) = (4\cdot 3^3 - 3\cdot 3)^2 = 99^2 = 9801$$

and since $C_k^2(3)$ must be such that $1+C_k^2(3) \ge 10^3$, we choose $k = 3$. Next, to find the poles of left half of the s–plane we first need to compute m, b, and c. From (11.65),

$$m = (\sqrt{1+\varepsilon^{-2}}+\varepsilon^{-1})^{1/k} = \left(\sqrt{1+\frac{1}{\varepsilon^2}}+\frac{1}{\sqrt{\varepsilon^2}}\right)^{1/3} = (\sqrt{2}+1)^{1/3}$$

or

$$m = 1.3415$$

and

$$m^{-1} = 0.7454$$

Then, from (11.63) and (11.64),

$$b = \frac{1.3415-0.7454}{2} = 0.298$$

$$c = \frac{1.3415+0.7454}{2} = 1.043$$

and the poles for $i = 0, 1,$ and 2 are found from (11.62), that is,

$$s_i = \omega_C\left[-b\sin\frac{(2i+1)\pi}{2k} + jc\cos\frac{(2i+1)\pi}{2k}\right]$$

Thus, the poles for this example are

$$s_0 = 5\left(-0.298\sin\frac{\pi}{6} + j1.043\cos\frac{\pi}{6}\right) = -0.745 + j4.516$$

$$s_1 = 5\left(-0.298\sin\frac{\pi}{2} + j1.043\cos\frac{\pi}{2}\right) = -1.49$$

$$s_2 = 5\left(-0.298\sin\frac{5\pi}{6} + j1.043\cos\frac{5\pi}{6}\right) = -0.745 - j4.516$$

Therefore, by substitution into (11.66) we obtain

$$G(s) = \frac{(-1)^3}{(s/s_0 - 1)(s/s_1 - 1)(s/s_2 - 1)} = \frac{-(-1.49)(-0.745 + j4.516)(-0.745 - j4.516)}{(s + 1.49)(s + 0.745 - j4.516)(s + 0.745 + j4.516)}$$

We will use the MATLAB script below to do these computations.

```
-(-1.49)*(-0.745+j*4.516)*(-0.745-j*4.516)

ans =
   31.2144

syms s; den=(s+1.49)*(s+0.745-j*4.516)*(s+0.745+j*4.516); simple(expand(den))

ans =
s^3+149/50*s^2+23169381/1000000*s+3121442869/100000000
```

Then,

$$G(s) = \frac{31.214}{s^3 + 2.980s^2 + 23.169s + 31.214} \tag{11.67}$$

To verify that the derived transfer function G(s) of (11.67) satisfies the filter specifications, we use the MATLAB script below to plot $|G(j\omega)|$.

```
w=0:0.01:100; s=j*w; Gs=31.214./(s.^3+2.98.*s.^2+23.169.*s+31.214);...
magGs=abs(Gs); dB=20.*log10(magGs); semilogx(w,dB);...
xlabel('Radian Frequency w rad/s – log scale'); ylabel('|G(w)| in dB');...
title('Magnitude of G(w) versus Radian Frequency'); grid
```

The plot is shown in Figure 11.25.

We can use the MATLAB **cheb1ap** function to design a Chebyshev Type I analog low–pass filter. Thus, the **[z,p,k] = cheb1ap(N,Rp)** statement where **N** denotes the order of the filter, returns the zeros, poles, and gain of an Nth order normalized prototype Chebyshev Type I analog low–pass filter with ripple **Rp** decibels in the pass band.

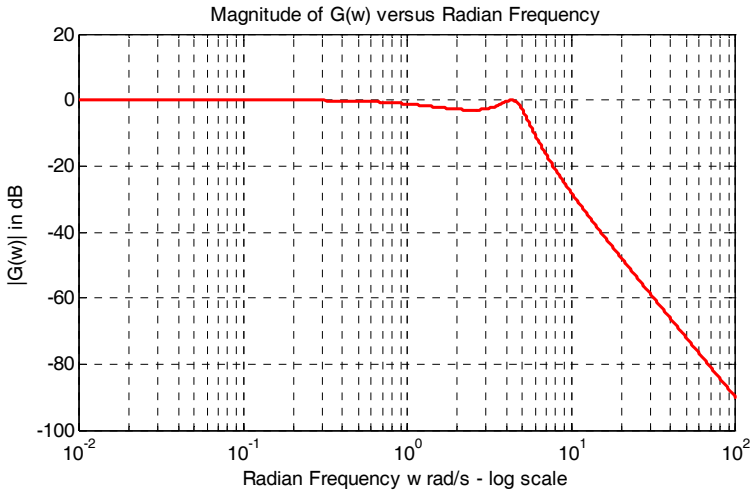


Figure 11.25. Plot for Example 11.8

Example 11.9

Use the MATLAB **cheb1ap** function to design a second–order Chebyshev Type I low–pass filter with 3 dB ripple in the pass–band.

Solution:

We use the script

```
w=0:0.05:400;                          % Define range to plot;...
[z,p,k]=cheb1ap(2,3);...
[b,a]=zp2tf(z,p,k);                    % Convert zeros and poles of G(s) to polynomial form;...
[mag,phase]=bode(b,a,w); hold on;...

b,a                                    % Display the b and a coefficients

b =
         0         0    0.5012

a =
    1.0000    0.6449    0.7079
```

Now, with the known values of a and b we use the **bode** function to produce the magnitude and phase Bode plots as follows:

```
bode(b,a), title('Bode Plot for Type 1 Chebyshev Low–Pass Filter'); grid
```

The Bode plots are shown in Figure 11.26.

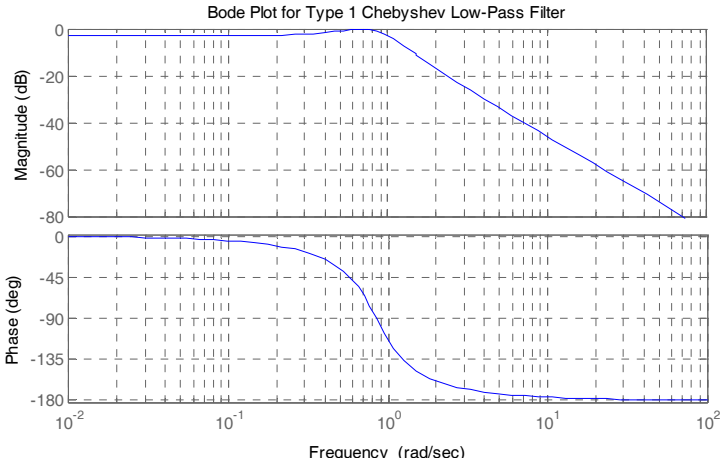


Figure 11.26. Bode plots for the filter of Example 11.9

On the Bode plots shown in Figure 11.26, the ripple is not so obvious. The reason is that this is a Bode plot with straight line approximations. To see the ripple, we use the MATLAB script below.

```
w=0:0.01:10; [z,p,k]=cheb1ap(2,3); [b,a]=zp2tf(z,p,k); Gs=freqs(b,a,w);...
xlabel('Frequency in rad/s'), ylabel('Magnitude of G(s) (absolute values)');...
semilogx(w,abs(Gs)); title('Type 1 Chebyshev Low–Pass Filter'); grid
```

The generated plot is shown in Figure 11.27.

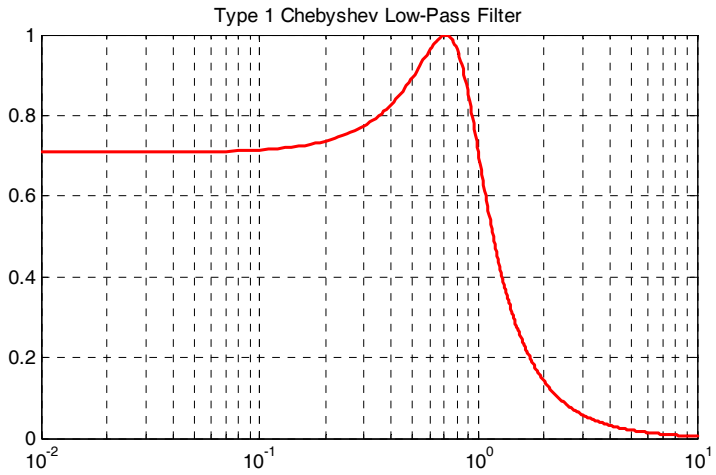


Figure 11.27. Magnitude characteristics for the Chebyshev Type I filter of Example 11.9

11.3.3 Chebyshev Type II Analog Low–Pass Filter Design

The *Chebyshev Type II*, also known as *Inverted Chebyshev* filters, are characterized by the following magnitude–square approximation.

$$A^2(\omega) = \frac{\varepsilon^2 C_k^2(\omega_C/\omega)}{1+\varepsilon^2 C_k^2(\omega_C/\omega)} \tag{11.68}$$

and has the ripple in the stop–band as opposed to Type I which has the ripple in the pass–band as shown in Figure 11.28.

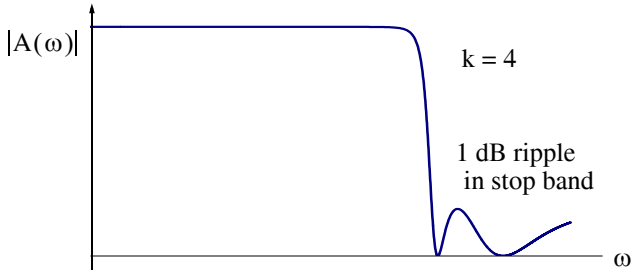


Figure 11.28. Chebyshev Type II low–pass filter

In relation (11.68), the frequency ω_C defines the beginning of the stop band.

We can design Chebyshev Type II low–pass filters with the MATLAB **cheb2ap** function. Thus, the statement **[z,p,k] = cheb2ap(N,Rs)** where **N** denotes the order of the filter, returns the zeros, poles, and gain of an Nth order normalized prototype Chebyshev Type II analog low–pass filter with ripple **Rs** decibels in the stop band.

Example 11.10

Using the MATLAB **cheb2ap** function, design a third order Chebyshev Type II analog filter with 3 dB ripple in the stop band.

Solution:

We begin with the MATLAB script below.

```
w=0:0.01:1000; [z,p,k]=cheb2ap(3,3); [b,a]=zp2tf(z,p,k); Gs=freqs(b,a,w);...
semilogx(w,abs(Gs)); xlabel('Frequency in rad/sec – log scale');...
ylabel('Magnitude of G(s) (absolute values)');...
title('Type 2 Chebyshev Low–Pass Filter, k=3, 3 dB ripple in stop band'); grid
```

The plot for this filter is shown in Figure 11.29.

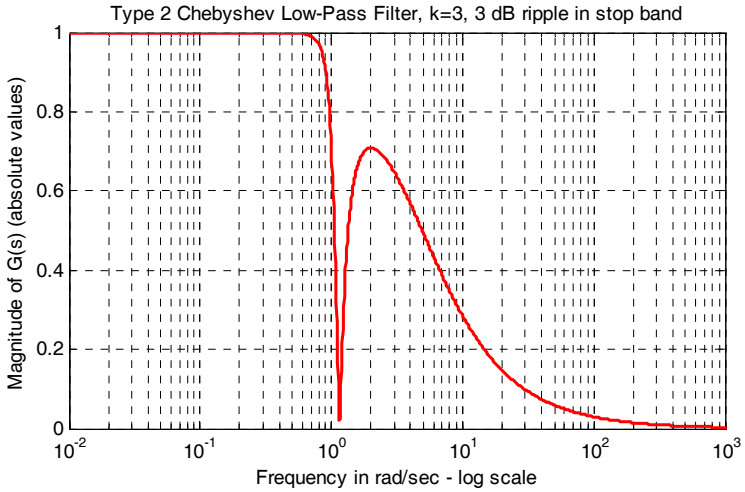


Figure 11.29. Plot for the Chebyshev Type II filter of Example 11.10

11.3.4 Elliptic Analog Low–Pass Filter Design

The *elliptic*, also known as *Cauer* filters, are characterized by the low–pass magnitude–squared function

$$A^2(\omega) = \frac{1}{1 + R_k^2(\omega/\omega_C)} \tag{11.69}$$

where $R_k(x)$ represents a rational elliptic function used with elliptic integrals. Elliptic filters have ripple in both the pass–band and the stop–band as shown in Figure 11.30.

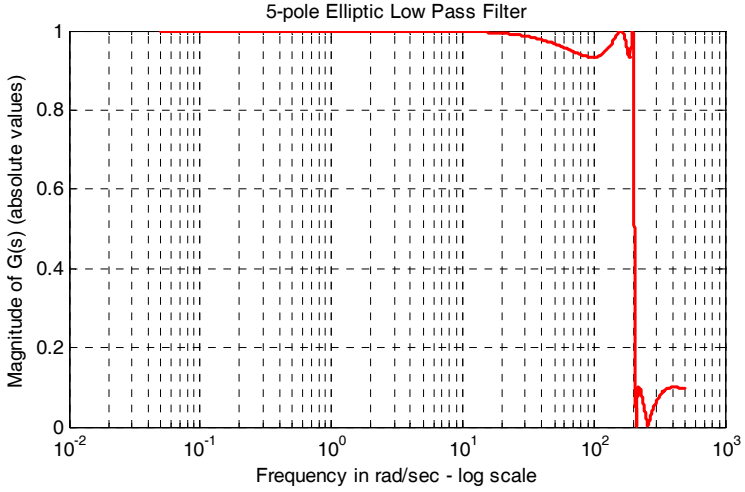


Figure 11.30. Characteristics of an elliptic low–pass filter

We can design elliptic low–pass filters with the MATLAB **ellip** function. The statement **[b,a] = ellip(N,Rp,Rs,Wn,'s')** where **N** is the order of the filter, designs an Nth order low–pass filter with ripple **Rp** decibels in the pass band, ripple **Rs** decibels in the stop band, **Wn** is the cutoff frequency, and **'s'** is used to specify analog elliptic filters. If **'s'** is not included in the above statement, MATLAB designs a digital filter. The plot of Figure 11.30 was obtained with the MATLAB script below:

```
w=0: 0.05: 500; [z,p,k]=ellip(5, 0.6, 20, 200, 's'); [b,a]=zp2tf(z,p,k);...
Gs=freqs(b,a,w); semilogx(w,abs(Gs));...
xlabel('Frequency in rad/sec – log scale'); ylabel('Magnitude of G(s) (absolute values)');...
title('5–pole Elliptic Low Pass Filter'); grid
```

Example 11.11

Use MATLAB to design a four–pole elliptic analog low–pass filter with 0.5 dB maximum ripple in the pass–band and 20 dB minimum attenuation in the stop–band with cutoff frequency at 200 rad/s.

Solution:

The solution is obtained with the following MATLAB script:

```
w=0: 0.05: 500; [z,p,k]=ellip(4, 0.5, 20, 200, 's'); [b,a]=zp2tf(z,p,k);...
Gs=freqs(b,a,w); semilogx(w,abs(Gs)); xlabel('Frequency in rad/sec – log scale');...
ylabel('Magnitude of G(s) (absolute values)'); title('4–pole Elliptic Low Pass Filter'); grid
```

The plot for this example is shown in Figure 11.31.

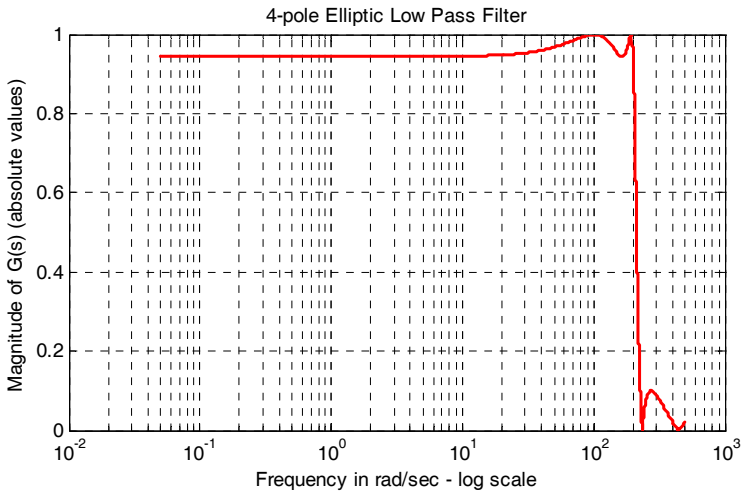


Figure 11.31. Plot for filter of Example 11.11

Next, suppose that we need to form the transfer function G(s) for this example. To do this, we

need to know the coefficients a_i and b_i of the denominator and numerator respectively, of G(s) in descending order. Because these are large numbers, we use the **format long** MATLAB command:

```
format long;...
a,b
```

and MATLAB displays

```
a =

  1.0e+009 *

   0.00000000100000   0.00000022702032   0.00007532236403

   0.00910982722080   1.15870252829421
b =

  1.0e+009 *

   0.00000000009998  -0.00000000000000   0.00002534545964

  -0.00000000000000   1.09388572421614
```

Thus, the transfer function for this filter is

$$G(s) = \frac{2.0487 \times 10^9}{s^4 + 339.793s^3 + 105866s^2 + 16.189 \times 10^6 + 2.072 \times 10^9} \tag{11.70}$$

11.4 High–Pass, Band–Pass, and Band–Elimination Filter Design

Transformation methods have been developed where a low–pass filter can be converted to another type of filter simply by transforming the complex variable s. These transformations are listed in Table 11.5 where ω_C is the cutoff frequency of a low–pass filter. The procedure is illustrated with the examples below.

Example 11.12

Compute the transfer function for a third–order band–pass Butterworth filter with 3 dB pass–band from 3 KHz to 5 KHz, from a third–order low–pass Butterworth filter with cutoff frequency $f_C = 1$ KHz.

Solution:

We first derive the transfer function for a third–order Butterworth low–pass filter with normalized frequency $\omega_C = 1$ rad/s. Using the MATLAB function **buttap** we write and execute the following script:

TABLE 11.5 Filter transformations

Analog Filter Frequency Transformations	
Filter Type, Frequency	Replace s in G(s) with
Low–Pass Filter, 3 dB pass–band, Normalized Frequency ω_C	No Change
Low–Pass Filter, 3 dB pass–band, Non–Normalized Frequency ω_{LP}	$\frac{s\omega_C}{\omega_{LP}}$
High–Pass Filter, 3 dB pass–band from $\omega = \omega_2$ to $\omega = \infty$	$\frac{\omega_{LP} \cdot \omega_2}{s}$
Band–Pass Filter, 3 dB pass–band from $\omega = \omega_{LP}$ to $\omega = \omega_2$	$\omega_C \cdot \frac{s^2 + \omega_{LP} \cdot \omega_2}{s(\omega_2 - \omega_{LP})}$
Band–Elimination Filter, 3 dB pass–band from $\omega = 0$ to $\omega = \omega_{LP}$, and from $\omega = \omega_2$ to $\omega = \infty$	$\omega_C \cdot \frac{s(\omega_2 - \omega_{LP})}{s^2 + \omega_{LP} \cdot \omega_2}$

format; [z, p, k]=buttap(3); [b,a]=zp2tf(z,p,k)

```
b =
     0     0     0     1
a =
    1.0000    2.0000    2.0000    1.0000
```

Thus, the transfer function for the third–order Butterworth low–pass filter with normalized cutoff frequency $\omega_c = 1$ rad/s is

$$G(s) = \frac{1}{s^3 + 2s^2 + 2s + 1} \tag{11.71}$$

Next, the actual cutoff frequency is given as $f_C = 1$ KHz or $\omega_C = 2\pi \times 10^3$ rad/s. Accordingly, in accordance with Table 11.5, we replace s with

$$\frac{s\omega_C}{\omega_{LP}} = \frac{s}{2\pi \times 10^3}$$

and we obtain

$$G\left(\frac{s}{2\pi \times 10^3}\right) = G'(s) = \frac{1}{(s/(2\pi \times 10^3))^3 + 2(s/(2\pi \times 10^3))^2 + 2(s/(2\pi \times 10^3)) + 1}$$

$$G'(s) = \frac{2.48 \times 10^{11}}{s^3 + 1.26 \times 10^4 s^2 + 7.89 \times 10^7 s + 2.48 \times 10^{11}} \tag{11.72}$$

Now, we replace s in the last expression of (11.72) with

$$\omega_C \cdot \frac{s^2 + \omega_{LP} \cdot \omega_2}{s(\omega_2 - \omega_{LP})} \tag{11.73}$$

or

$$1 \cdot \frac{s^2 + 2\pi \times 10^3 \times 3 \times 2\pi \times 10^3}{s(3 \times 2\pi \times 10^3 - 2\pi \times 10^3)} = \frac{s^2 + 12 \times \pi^2 \times 10^6}{s(4\pi \times 10^3)} = \frac{s^2 + 1.844 \times 10^8}{1.257 \times 10^4 s}$$

Then,

$$G''(s) = \frac{2.48 \times 10^{11}}{\left(\frac{s^2 + 1.844 \times 10^8}{1.257 \times 10^4 s}\right)^3 + \left(\frac{s^2 + 1.844 \times 10^8}{1.257 \times 10^4 s}\right)^2 + \frac{s^2 + 1.844 \times 10^8}{1.257 \times 10^4 s} + 2.48 \times 10^{11}}$$

We see that the computations, using the transformations of Table 11.5 become quite tedious. Fortunately, we can use the MATLAB **lp2lp**, **lp2hp**, **lp2bp**, and **lp2bs** functions to transform a low pass filter with normalized cutoff frequency, to another low–pass filter with any other specified frequency, or to a high–pass filter, or to a band–pass filter, or to a band–elimination filter respectively.

Example 11.13

Use the MATLAB **buttap** and **lp2lp** functions to derive the transfer function of a third–order Butterworth low–pass filter with cutoff frequency $f_C = 2$ KHz.

Solution:

We will use the **buttap** command to derive the transfer function $G(s)$ of the filter with normalized cutoff frequency at $\omega_C = 1$ rad/s. Then, we will use the command **lp2lp** to transform $G(s)$ to $G'(s)$ with cutoff frequency at $f_C = 2$ KHz, or $\omega_C = 2\pi \times 2 \times 10^3$ rad/s.

```
format short e                % Will be used to find coefficients for transfer function;...
%  Design 3 pole Butterworth low–pass filter (wcn=1 rad/s);...
[z,p,k]=buttap(3);            % To find transfer function with normalized cutoff frequency;...
[b,a]=zp2tf(z,p,k);           % Compute num, den coefficients of this filter (wcn=1rad/s);...
f=100:100:10000;              % Define frequency range to plot;...
w=2*pi*f;                     % Convert to rads/sec;...
fc=2000;                      % Define actual cutoff frequency at 2 KHz;...
```

```
wc=2*pi*fc;                 % Convert desired cutoff frequency to rads/sec;...
[bn,an]=lp2lp(b,a,wc);      % Compute num, den of filter with fc = 2 kHz;...
Gsn=freqs(bn,an,w);         % Compute transfer function of filter with fc = 2 kHz;...
semilogx(w,20.*log10(abs(Gsn))); xlabel('Radian Frequency w (rad/sec) – log scale'),...
ylabel('Magnitude of Transfer Function (dB)'),...
title('3–pole Butterworth low–pass filter with fc=2 kHz or wc = 12.57 kr/s'); grid
```

The plot for the magnitude of this transfer function is shown in Figure 11.32.

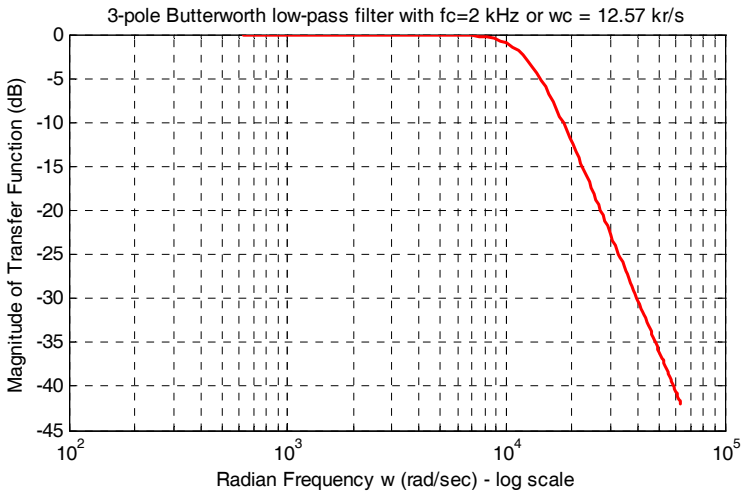


Figure 11.32. Magnitude for the transfer function for Example 11.13

The coefficients of the numerator and denominator of the transfer function are as follows:

```
b, a, bn, an
```

```
b =
             0             0             0   1.0000e+000
a =
  1.0000e+000   2.0000e+000   2.0000e+000   1.0000e+000
bn =
  1.9844e+012
an =
  1.0000e+000   2.5133e+004   3.1583e+008   1.9844e+012
```

Thus, the transfer function with normalized cutoff frequency $\omega_{Cn} = 1$ rad/s is

$$G(s) = \frac{1}{s^3 + 2s^2 + 2s + 1} \tag{11.74}$$

and with actual cutoff frequency $\omega_{Cn} = 2\pi \times 2000 \text{ rad/s} = 1.2566 \times 10^4$ is

$$G'(s) = \frac{1.9844 \times 10^{12}}{s^3 + 2.5133 \times 10^4 s^2 + 3.1583 \times 10^8 s + 1.9844 \times 10^{12}} \tag{11.75}$$

Example 11.14

Use the MATLAB commands **cheb1ap** and **lp2hp** to derive the transfer function of a 3–pole Chebyshev Type I high–pass analog filter with cutoff frequency $f_C = 5$ KHz.

Solution:

We will use the **cheb1ap** command to derive the transfer function G(s) of the low–pass filter with normalized cutoff frequency at $\omega_C = 1$ rad/s. Then, we will use the command **lp2hp** to transform G(s) to another transfer function G'(s) with cutoff frequency at $f_C = 5$ KHz or $\omega_C = 2\pi \times 5 \times 10^3$ rad/s

```
% Design 3 pole Type 1 Chebyshev low–pass filter, wcn=1 rad/s;...
[z,p,k]=cheb1ap(3,3);...
[b,a]=zp2tf(z,p,k);              % Compute num, den coef. with wcn=1 rad/s;...
f=1000:100:100000;               % Define frequency range to plot;...
fc=5000;                         % Define actual cutoff frequency at 5 KHz;...
wc=2*pi*fc;                      % Convert desired cutoff frequency to rads/sec;...
[bn,an]=lp2hp(b,a,wc);           % Compute num, den of high–pass filter with fc = 5 KHz;...
Gsn=freqs(bn,an,2*pi*f);         % Compute and plot transfer function of filter with fc = 5 KHz;...
semilogx(f,20.*log10(abs(Gsn)));...
xlabel('Frequency (Hz) – log scale'); ylabel('Magnitude of Transfer Function (dB)');...
title('3–pole Type 1 Chebyshev high–pass filter with fc=5 KHz '); grid
```

The magnitude of this transfer function is plotted as shown in Figure 11.33.

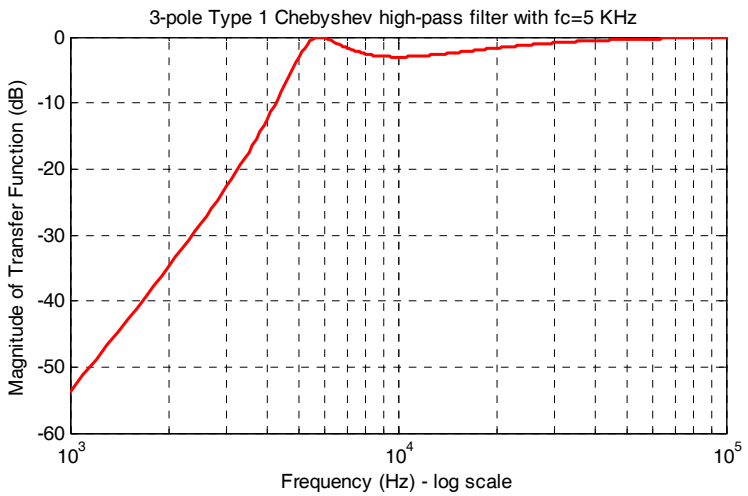


Figure 11.33. Magnitude of the transfer for Example 11.14

```
b, a, bn, an
```

The coefficients of the numerator and denominator of the transfer function are as follows:

```
b =
   0              0              0   2.5059e-001

a =
   1.0000e+000   5.9724e-001   9.2835e-001   2.5059e-001

bn =
   1.0000e+000   2.2496e-011  -1.4346e-002  -6.8973e-003

an =
   1.0000e+000   1.1638e+005   2.3522e+009   1.2373e+014
```

Therefore, the transfer function with normalized cutoff frequency $\omega_{Cn} = 1$ rad/s is

$$G(s) = \frac{0.2506}{s^3 + 0.5972s^2 + 0.9284s + 0.2506} \tag{11.76}$$

and with actual cutoff frequency $\omega_{Cn} = 2\pi \times 5000 \text{ rad/s} = 3.1416 \times 10^4$, is

$$G'(s) = \frac{s^3}{s^3 + 1.1638 \times 10^5 s^2 + 2.3522 \times 10^9 s + 1.2373 \times 10^{14}} \tag{11.77}$$

Example 11.15

Use the MATLAB functions **buttap** and **lp2bp** to derive the transfer function of a 3–pole Butterworth analog band–pass filter with the pass band frequency centered at $f_0 = 4$ KHz, and bandwidth BW = 2 KHz.

Solution:

We will use the **buttap** function to derive the transfer function G(s) of the low–pass filter with normalized cutoff frequency at $\omega_C = 1$ rad/s. We found this transfer function in Example 11.12 as given by (11.71), Page 11–42. However, to maintain a similar MATLAB script as in the previous examples, we will include it in the script below. Then, we will use the command **lp2bp** to transform G(s) to another transfer function G'(s) with centered frequency at $f_0 = 4$ KHz or $\omega_0 = 2\pi \times 4 \times 10^3$ rad/s, and bandwidth BW = 2 KHz or BW = $2\pi \times 2 \times 10^3$ rad/s

```
format short e;...
[z,p,k]=buttap(3);        % Design 3 pole Butterworth low-pass filter with wcn=1 rad/s;...
[b,a]=zp2tf(z,p,k);       % Compute numerator and denominator coefficients for wcn=1 rad/s;...
f=100:100:100000;         % Define frequency range to plot;...
f0=4000;                  % Define centered frequency at 4 KHz;...
W0=2*pi*f0;               % Convert desired centered frequency to rads/sec;...
fbw=2000;                 % Define bandwidth;...
Bw=2*pi*fbw;              % Convert desired bandwidth to rads/sec;...
```

```
[bn,an]=lp2bp(b,a,W0,Bw);% Compute num, den of band–pass filter;...
% Compute and plot the magnitude of the transfer function of the band–pass filter;...
Gsn=freqs(bn,an,2*pi*f); semilogx(f,20.*log10(abs(Gsn)));...
xlabel('Frequency f (Hz) – log scale'); ylabel('Magnitude of Transfer Function (dB)');...
title('3–pole Butterworth band–pass filter with f0 = 4 KHz, BW = 2KHz'); grid
```

The plot for this band–pass filter is shown in Figure 11.34.

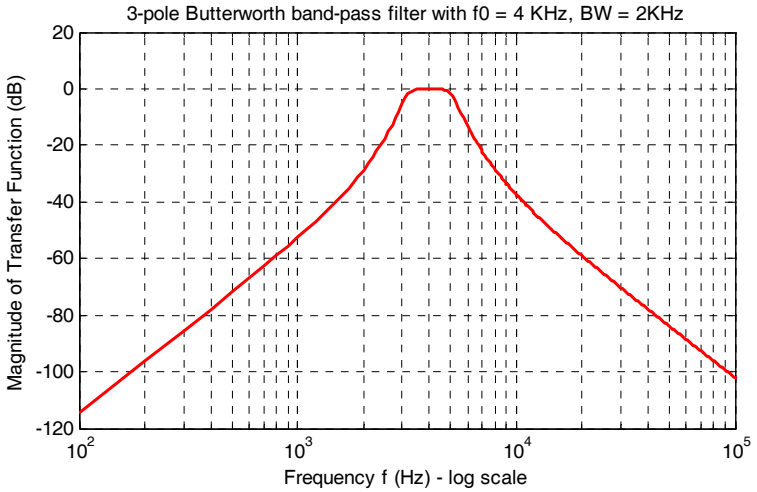


Figure 11.34. Plot for the band-pass filter of Example 11.15

bn, an

The coefficients b_n and a_n are as follows:

```
bn =
  1.9844e+012  -4.6156e+001  -1.6501e+005  -2.5456e+009

an =
  1.0000e+000  2.5133e+004  2.2108e+009  3.3735e+013  1.3965e+018
  1.0028e+022  2.5202e+026
```

Since the numerator b_n and denominator a_n coefficients are too large to be written in a one line equation, we have listed them in tabular form as shown below.

Power of s	Numerator b_n	Denominator a_n
s^6	0	1
s^5	0	2.5133×10^4
s^4	0	2.2108×10^9
s^3	1.9844×10^{12}	3.3735×10^{13}
s^2	-4.6156×10^1	1.3965×10^{18}
s	-1.6501×10^5	1.0028×10^{22}
Constant	-2.5456×10^9	2.5202×10^{26}

Example 11.16

Use the MATLAB functions **buttap** and **lp2bs** to derive the transfer function of a 3–pole Butterworth band–elimination (band–stop) filter with the stop band frequency centered at $f_0 = 5$ KHz, and bandwidth BW = 2 KHz.

Solution:

We will use the **buttap** function to derive the transfer function G(s) of the low–pass filter with normalized cutoff frequency at $\omega_C = 1$ rad/s. We found this transfer function as (11.71) in Example 11.12, Page 11–42. However, to maintain a similar MATLAB script as in the previous examples, we will include it in the script which follows. Accordingly, we will use the **lp2bs** function to transform G(s) to another transfer function G'(s) with centered frequency at $f_0 = 5$ KHz, or radian frequency $\omega_0 = 2\pi \times 5 \times 10^3$ rad/s, and bandwidth BW = 2 KHz or BW $= 2\pi \times 2 \times 10^3$ rad/s.

```
[z,p,k]=buttap(3);              % Design 3–pole Butterworth low–pass filter, wcn = 1 r/s;...
[b,a]=zp2tf(z,p,k);             % Compute num, den coefficients of this filter, wcn=1 r/s;...
f=1000:100:10000;               % Define frequency range to plot;...
f0=5000;                        % Define centered frequency at 5 kHz;...
W0=2*pi*f0;                     % Convert centered frequency to r/s;...
fbw=2000;                       % Define bandwidth;...
Bw=2*pi*fbw;                    % Convert bandwidth to rad/s;...
% Compute numerator and denominator coefficients of desired band stop filter;...
[bn,an]=lp2bs(b,a,W0,Bw);...
% Compute and plot magnitude of the transfer function of the band stop filter;...
Gsn=freqs(bn,an,2*pi*f); semilogx(f,20.*log10(abs(Gsn)));...
xlabel('Frequency in Hz – log scale'); ylabel('Magnitude of Transfer Function (dB)');...
title('3-pole Butterworth band-elimination filter with f0=5 KHz, BW = 2 KHz'); grid
```

The magnitude response for this band–elimination filter is shown in Figure 11.35.

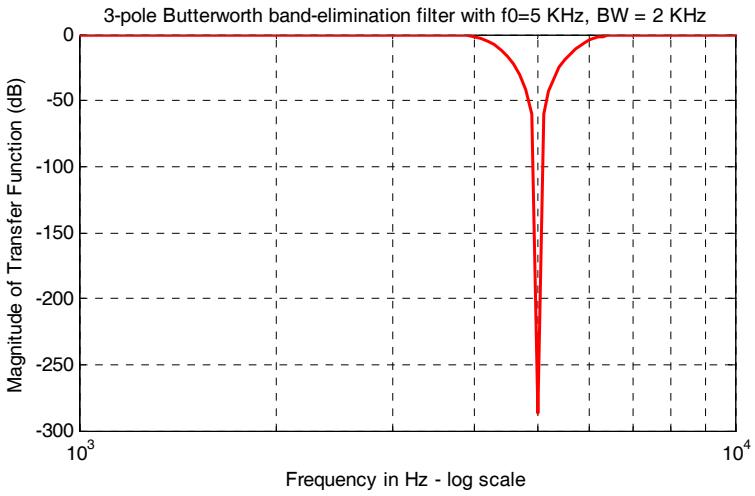


Figure 11.35. Magnitude response for the band–elimination filter of Example 11.16

The coefficients b_n and a_n are as follows:

bn, an

```
bn =
  1.0000e+000   -7.6352e-012   2.9609e+009   -1.5071e-002
  2.9223e+018   -7.4374e+006   9.6139e+026

an =
  1.0000e+000    2.5133e+004   3.2767e+009    5.1594e+013
  3.2340e+018    2.4482e+022   9.6139e+026
```

As in the previous example, we list the numerator b_n and denominator a_n coefficients in tabular form as shown below.

Power of s	Numerator b_n	Denominator a_n
s^6	1	1
s^5	-7.6352×10^{-12}	2.5133×10^{4}
s^4	2.9609×10^{-6}	3.2767×10^{9}
s^3	-1.5071×10^{-2}	5.1594×10^{13}
s^2	2.9223×10^{18}	3.2340×10^{18}
s	-7.4374×10^{6}	2.4482×10^{22}
Constant	9.6139×10^{26}	9.6139×10^{26}

In all of the above examples, we have shown the magnitude, but not the phase response of each filter type. However, we can use the MATLAB function **bode(num,den)** to generate both the magnitude and phase responses of any transfer function describing the filter type, as shown by the following example.

Example 11.17

Use the MATLAB **bode** function to plot the magnitude and phase characteristics of the 3–pole Butterworth low-pass filter with unity gain and normalized frequency at $\omega_C = 1$ rad/s.

Solution:

We know, from Example 11.12, Page 11–42, that the transfer function for this type of filter is

$$G(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}$$

We can obtain the magnitude and phase characteristics with the following MATLAB script:

```
num=[0  0  0  1]; den=[1  2  2  1]; bode(num,den),...
title('Bode Plot for 3–pole Butterworth Low–Pass Filter'); grid
```

The magnitude and phase characteristics are shown in Figure 11.36.

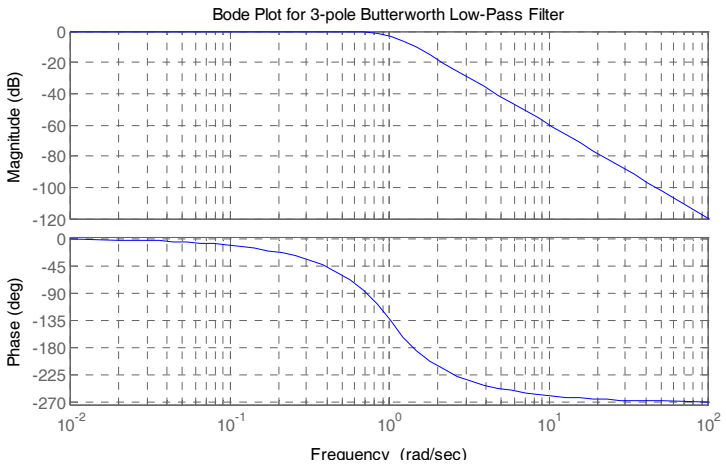


Figure 11.36. Bode plots for 3–pole Butterworth low–pass filter, Example 11.17

We conclude the discussion on analog filters with Table 11.6 listing the advantages and disadvantages of each type.

TABLE 11.6 Advantages / Disadvantages of different types of filters

Filter Type	Advantages	Disadvantages
Butterworth	• Simplest design • Flat pass band	• Slow rate of attenuation for order 4 or less
Chebyshev Type 1	• Sharp cutoff rate in transition (pass to stop) band	• Ripple in pass band • Bad (non–linear) phase response
Chebyshev Type II	• Sharp cutoff rate in transition (pass to stop) band	• Ripple in stop band • Bad (non–linear) phase response
Elliptic (Cauer)	• Sharpest cutoff rate among all other types of filters	• Ripple in both pass band and stop band • Worst (most non–linear) phase response among the other types of filters.

11.5 Digital Filters

A *digital filter* is essentially a computational process (algorithm) that converts one sequence of numbers x[n] representing the input, to another sequence y[n] that represents the output. Thus, a digital filter, in addition of filtering out desired bands of frequency, can also be used as a computational means of performing other functions such as integration, differentiation, and estimation.

The input–output difference equation that relates the output to the input can be expressed in the discrete time domain as a summation of the form

$$y[n] = \sum_{i=0}^{k} a_i x[n-i] - \sum_{i=0}^{k} b_i y[n-i] \tag{11.78}$$

or, in the z–domain as

$$G(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{k} a_i z^{-i}}{1 + \sum_{i=0}^{k} b_i z^{-i}} \tag{11.79}$$

Therefore, the design of a digital filter to perform a desired function, entails the determination of the coefficients a_i and b_i.

Digital filters are classified in terms of the duration of the impulse response, and in forms of realization.

1. Impulse Response Duration

a. An *Infinite Impulse Response* (IIR) digital filter has infinite number of samples in its impulse

response h[n]

b. A *Finite Impulse Response* (FIR) digital filter has a finite number of samples in its impulse response h[n]

2. Realization

a. In a *Recursive Realization* digital filter the output is dependent on the input and the *previous* values of the output. In a recursive digital filter, both the coefficients a_i and b_i are present.

b. In a *Non–Recursive Realization* digital filter the output depends on present and past values of the input only. In a non–recursive digital filter, only the coefficients a_i are present, i.e., $b_i = 0$.

Figure 11.37 shows third–order (3–delay element) recursive realization and Figure 11.38 shows a third–order non–recursive realization. The components of either realization are shown in Figure 11.39. Generally, IIR filters are implemented by recursive realization, whereas FIR filters are implemented by non–recursive realization.

Filter design methods have been established, and prototype circuits have been published. Thus, we can choose the appropriate prototype to satisfy the requirements. Transformation methods are also available to map an analog prototype to an equivalent digital filter. Three well known transformation methods are the following:

1. The *Impulse Invariant Method* which produces a digital filter whose impulse response consists of the sampled values of the impulse response of an analog filter.
2. The *Step Invariant Method* which produces a digital filter whose step response consists of the sampled values of the step response of an analog filter.

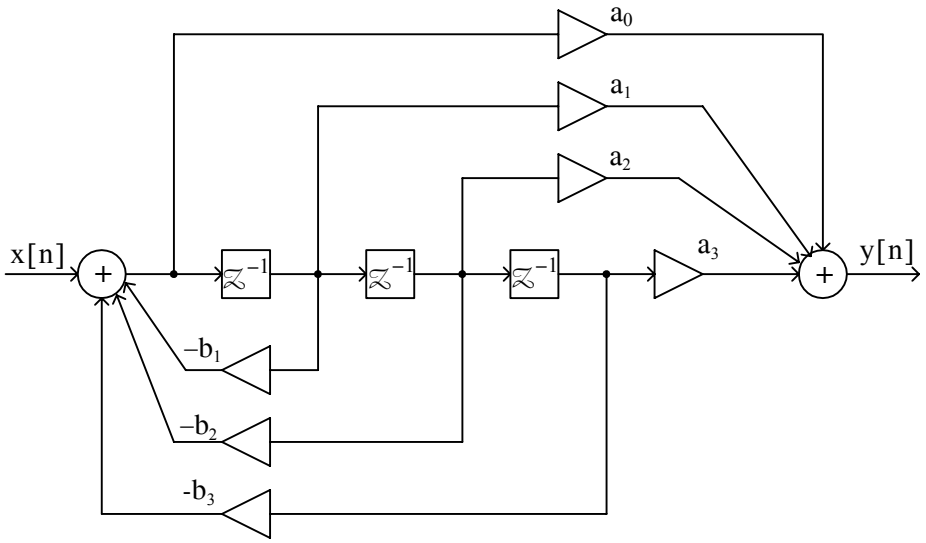


Figure 11.37. Recursive digital filter realization

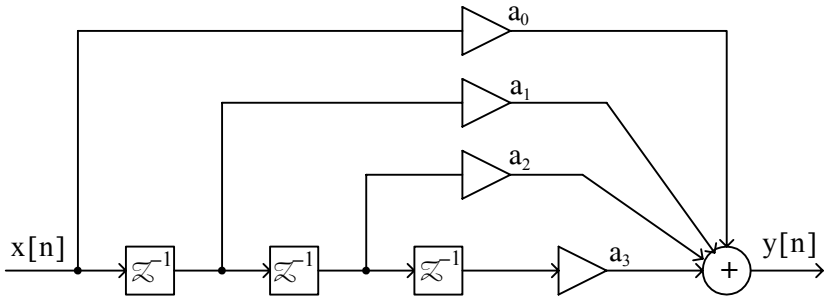


Figure 11.38. Non–recursive digital filter realization

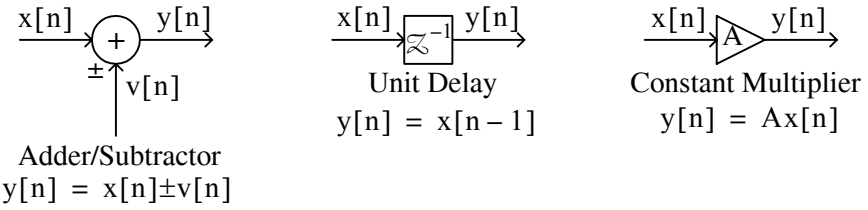


Figure 11.39. Components of recursive and non–recursive digital filter realization

3. The *Bilinear Transformation* which uses the transformation

$$s = \frac{2}{T_s} \cdot \frac{z-1}{z+1} \; * \tag{11.80}$$

to transform the left–half of the s–plane into the interior of the unit circle in the z–plane.

We will discuss only the bilinear transformation.

We recall from (9.67) of Chapter 9, Page 9–22, that

$$F(z) = G(z) = G(s)\Big|_{s = \frac{1}{T_S}\ln z} \tag{11.81}$$

But the relation $s = \frac{1}{T_S}\ln z$ is a multi–valued transformation and, as such, cannot be used to derive a rational form in z. It can be approximated as

$$s = \frac{1}{T_S}\ln z = \frac{2}{T}\left[\frac{z-1}{z+1} + \frac{1}{3}\left(\frac{z-1}{z+1}\right)^3 + \frac{1}{5}\left(\frac{z-1}{z+1}\right)^5 + \frac{1}{7}\left(\frac{z-1}{z+1}\right)^7 + \ldots\right] \approx \frac{2}{T_S} \cdot \frac{z-1}{z+1} \tag{11.82}$$

Substitution (11.82) into (11.81) yields

* T_s *is the sampling period, that is, the reciprocal of the sampling frequency* f_s

$$G(z) = G(s)\Big|_{s=\frac{2}{T_S}\cdot\frac{z-1}{z+1}} \tag{11.83}$$

and with the substitution $z = e^{j\omega_d T_S}$, we obtain

$$G(e^{j\omega_d T_S}) = G\left(\frac{2}{T_S}\cdot\frac{e^{j\omega_d T_S}-1}{e^{j\omega_d T_S}+1}\right) \tag{11.84}$$

Since the $z \to s$ transformation maps the unit circle into the $j\omega$ axis on the s–plane, the quantity $\frac{2}{T_s}\cdot\frac{e^{j\omega_d}-1}{e^{j\omega_d}+1}$ and $j\omega$ must be equal to some point $\omega = \omega_a$ on the $j\omega$ axis, that is,

$$j\omega_a = \frac{2}{T_S}\cdot\frac{e^{j\omega_d T_S}-1}{e^{j\omega_d T_S}+1}$$

or

$$\omega_a = \frac{1}{j}\cdot\frac{2}{T_s}\cdot\frac{e^{j\omega_d T_S}-1}{e^{j\omega_d T_S}+1} = \frac{2}{T_S}\cdot\frac{1/(j2)}{1/2}\cdot\frac{e^{j\omega_d T_S/2}-e^{-j\omega_d T_S/2}}{e^{j\omega_d T_S/2}+e^{-j\omega_d T_S/2}} = \frac{2}{T_S}\frac{\sin(\omega_d T_S)/2}{\cos(\omega_d T_S)/2}$$

or

$$\boxed{\omega_a = \frac{2}{T_S}\cdot\tan\frac{\omega_d T_S}{2}} \tag{11.85}$$

We see that the analog frequency to digital frequency transformation results in a non–linear mapping; this condition is known as *warping*. For instance, the frequency range $0 \le \omega_a \le \infty$ in the analog frequency is warped into the frequency range $0 \le \omega_d \le \pi/T_s$ in the digital frequency.

To express ω_d in terms of ω_a, we rewrite (11.85) as

$$\tan\frac{\omega_d T_S}{2} = \frac{\omega_a T_S}{2}$$

Then,

$$\omega_d T_S = 2\tan^{-1}\frac{\omega_a T_S}{2}$$

and for small $\omega_a T_s/2$,

$$\tan^{-1}\frac{\omega_a T_S}{2} \approx \frac{\omega_a T_S}{2}$$

Therefore,

$$\omega_d T_S \approx 2\frac{\omega_a T_S}{2} \approx \omega_a T_S \tag{11.86}$$

that is, for small frequencies,

$$\omega_d \approx \omega_a \tag{11.87}$$

In MATLAB, z is a function of normalized frequency and thus the range of frequencies in G(z) is from 0 to π. Then (11.86), when used with MATLAB, becomes

$$\boxed{\omega_d \approx \frac{\omega_a T_S}{\pi}} \tag{11.88}$$

The effect of warping can be eliminated by *pre–warping* the analog filter prior to application of the bilinear transformation. This is accomplished with the use of (11.85).

Example 11.18

Compute the transfer function G(z) of a low–pass digital filter with 3 dB cutoff frequency at 20 Hz, and attenuation of at least 10 dB for frequencies greater than 40 Hz. The sampling frequency f_s is 200 Hz. Compare the magnitude plot with that obtained by a low–pass analog filter with the same specifications.

Solution:

We will apply the bilinear transformation, and using the procedure of Example 11.5, Page 11–20, we arbitrarily choose a second order Butterworth low–pass filter which, as we see from the curves of Figure 11.17, Page 11–19, meets the stop–band specification.

The transfer function G(s) of the analog low–pass filter with normalized frequency at $\omega_C = 1$ rad/s is found with the MATLAB **buttap** function as follows:

```
[z,p,k]=buttap(2); [b,a] = zp2tf(z,p,k)

b =
     0     0     1
a =
    1.0000    1.4142    1.0000
```

Thus, the transfer function with normalized frequency, denoted as $G_n(s)$, is

$$G_n(s) = \frac{1}{s^2 + 1.414s + 1} \tag{11.89}$$

Now, we must transform this transfer function to another with the actual cutoff frequency at 20 Hz. We denote it as $G_a(s)$.

We will first pre–warp the analog frequency which, by relation (11.85), Page 11–54, is related to the digital frequency as

$$\omega_a = \frac{2}{T_S} \cdot \tan\frac{\omega_d T_S}{2}$$

where

$$T_S = \frac{1}{f_S} = \frac{1}{200}$$

Denoting the analog cutoff (3 dB) frequency as ω_{ac}, we obtain

$$\omega_{ac} = 400\tan\frac{2\pi(20)/200}{2} = 400\tan(0.1\pi) \approx 130 \text{ rad/s}$$

or

$$f_{ac} = \frac{130}{2\pi} \approx 20.69 \text{ Hz}$$

As expected from relation (11.87), Page 11–55, this frequency is very close to the the discrete–time frequency $f_{dc} = 20$ Hz, and thus from (11.89),

$$G_a(s) \approx G_n(s) = \frac{1}{s^2 + 1.414s + 1} \tag{11.90}$$

Relation (11.90) applies only when the cutoff frequency is normalized to to $\omega_C = 1$ rad/s. If $\omega_C \neq 1$, we must scale the transfer function in accordance with relation (11.34), Page 11–19, that is,

$$G(s)_{actual} = G\left(\frac{s}{\omega_{actual}}\right)$$

For this example, $\omega_{actual} = 130$ rad/s, and thus we replace s with $s/130$ and we obtain

$$G_a(s) = \frac{1}{(s/130)^2 + 1.414s/130 + 1}$$

We will use MATLAB to simplify this expression.

```
syms s; simplify(1/((s/130)^2+1.414*s/130+1))
ans =
  845000/(50*s^2+9191*s+845000)
845000/50
ans =
   16900
9191/50
ans =
```

```
183.8200
```

Then,

$$G_a(s) = \frac{845000}{50s^2 + 9191s + 845000} = \frac{16900}{s^2 + 183.82s + 16900} \tag{11.91}$$

and making the substitution of $s = \frac{2}{T_S} \cdot \frac{z-1}{z+1} = 400 \cdot \frac{z-1}{z+1}$, we obtain

$$G(z) = \frac{16900}{\left(400 \cdot \frac{z-1}{z+1}\right)^2 + \frac{183.82 \times 400(z-1)}{(z+1)} + 16900}$$

We use the MATLAB script below to simplify this expression.

```
syms z; simplify(16900/((400*(z–1)/(z+1))^2+183.82*400*(z–1)/(z+1)+16900))
ans =
  4225*(z+1)^2/(62607*z^2–71550*z+25843)
expand(4225*(z+1)^2)
ans =
  4225*z^2+8450*z+4225
```

and thus

$$G(z) = \frac{4225z^2 + 8450z + 4225}{62607z^2 - 71550z + 25843} \tag{11.92}$$

We will use the MATLAB **freqz** function to plot the magnitude of $G(z)$, but we must first express it in negative powers of z. Dividing each term of (11.92) by $62607z^2$, we obtain

$$G(z) = \frac{0.0675 + 0.1350z^{-1} + 0.0675z^{-2}}{1 - 1.1428z^{-1} + 0.4128z^{-2}} \tag{11.93}$$

The MATLAB script below will generate $G(z)$ and will plot the magnitude of this transfer function.

```
bz=[0.0675  0.1350  0.0675]; az=[1  –1.1428  0.4128]; [Gz, wT]=freqz(bz,az,20,200);...
semilogx(fs,20.*log10(abs(Gz))); xlabel('Frequency in Hz – log scale');...
ylabel('Magnitude (dB)'); title('Digital Low–Pass Filter, Example 11.18'); grid
```

The magnitude is shown on the plot of Figure 11.40.

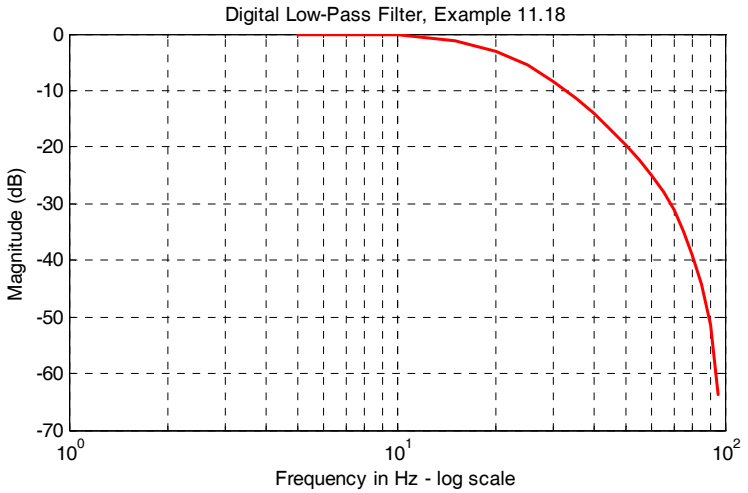


Figure 11.40. Frequency response for the digital low–pass filter of Example 11.18

Let us now plot the analog equivalent to compare the digital to the analog frequency response. The MATLAB script below produces the desired plot.

```
[z,p,k]=buttap(2); [b,a]=zp2tf(z,p,k); f=1:1:100; fc=20; [bn,an]=lp2lp(b,a,fc);...
Gs=freqs(bn,an,f);...
semilogx(f,20.*log10(abs(Gs))); xlabel('Frequency in Hz log scale'), ylabel('Magnitude (dB)');...
title('Analog Low–Pass Filter, Example 11.18'); grid
```

The frequency response for the analog low–pass equivalent is shown in Figure 11.41.

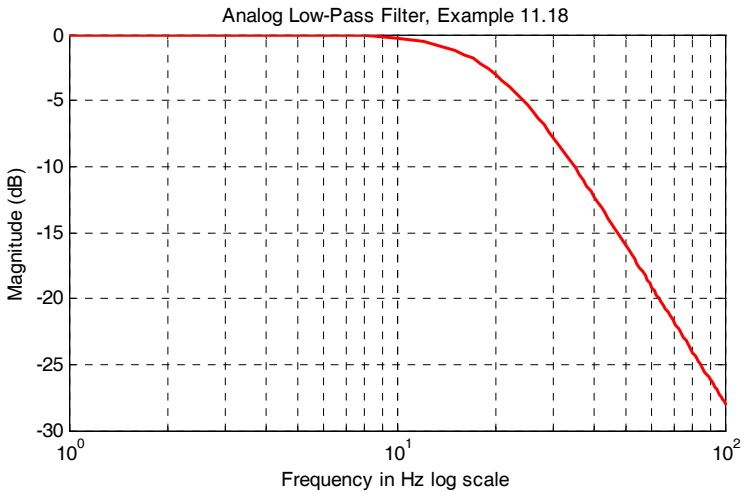


Figure 11.41. Frequency response for analog low–pass filter equivalent, Example 11.18

Comparing the digital filter plot of Figure 11.40 with its equivalent the analog filter of Figure 11.41, we observe that the magnitude is greater than0 –3 dB for frequencies less than 20Hz, and is smaller than (-10 dB) for frequencies larger than 40Hz. Therefore, both the digital and analog low–pass filters meet the specified requirements.

An analog filter transfer function can be mapped to a digital filter transfer function directly with the MATLAB **bilinear** function. The procedure is illustrated with the following example.

Example 11.19

Use the MATLAB **bilinear** function to derive the low–pass digital filter transfer function G(z) from a second-order Butterworth analog filter with a 3 dB cutoff frequency at 50 Hz, and sampling rate $f_S = 500$ Hz.

Solution:

We will use the following MATLAB script to produce the desired digital filter transfer function.

```
[z,p,k]=buttap(2); [num,den]=zp2tf(z,p,k); wc=2*pi*50;...
[num1,den1]=lp2lp(num,den,wc); T=1/500; [numd,dend]=bilinear(num1,den1,1/T)
```

```
numd =
    0.0640    0.1279    0.0640
dend =
    1.0000   -1.1683    0.4241
```

Therefore, the transfer function G(z) for this filter is

$$G(z) = \frac{0.0640z^2 + 0.1279z + 0.0640}{z^2 - 1.1683z + 0.4241} \tag{11.94}$$

MATLAB provides us with all the functions that we need to design digital filters using analog prototypes. These are listed below with the indicated notations.

N = order of the filter
Wn = normalized cutoff frequency
Rp = pass band ripple
Rs = stop band ripple
B = B(z), i.e., the numerator of the discrete transfer function G(z) = B(z)/A(z)
A = A(z), i.e., the denominator of the discrete transfer function G(z)

For Low–Pass Filters

```
[B,A] = butter(N,Wn)
[B,A] = cheb1(N,Rp,Wn)
[B,A] = cheb2(N,Rs,Wn)
[B,A] = ellip(N,Rp,Rs,Wn)
```

For High–Pass Filters

```
[B,A] = butter(N,Wn,'high')
[B,A] = cheb1(N,Rp,Wn,'high')
[B,A] = cheb2(N,Rs,Wn,'high')
[B,A] = ellip(N,Rp,Rs,Wn,'high')
```

Band–Pass Filters

```
[B,A] = butter(N,[Wn1,Wn2])
[B,A] = cheb1(N,Rp,[Wn1,Wn2])
[B,A] = cheb2(N,Rs,[Wn1,Wn2])
[B,A] = ellip(N,Rp,Rs,[Wn1,Wn2])
```

Band–Elimination Filters

```
[B,A] = butter(N,[Wn1,Wn2],'stop')
[B,A] = cheb1(N,Rp,[Wn1,Wn2],'stop')
[B,A] = cheb2(N,Rs,[Wn1,Wn2],'stop')
[B,A] = ellip(N,Rp,Rs,[Wn1,Wn2],'stop')
```

Example 11.20

The transfer functions of (11.95) through (11.98) below, describe different types of digital filters. Use the MATLAB **freqz** command to plot the magnitude versus radian frequency.

$$G_1(z) = \frac{(2.8982 + 8.6946z^{-1} + 8.6946z^{-2} + 2.8982z^{-3}) \cdot 10^{-3}}{1 - 2.3741z^{-1} + 1.9294z^{-2} - 0.5321z^{-3}} \tag{11.95}$$

$$G_2(z) = \frac{0.5276 - 1.5828z^{-1} + 1.5828z^{-2} - 0.5276z^{-3}}{1 - 1.7600z^{-1} + 1.1829z^{-2} - 0.2781z^{-3}} \tag{11.96}$$

$$G_3(z) = \frac{(6.8482 - 13.6964z^{-2} + 6.8482z^{-4}) \cdot 10^{-4}}{1 + 3.2033z^{-1} + 4.5244z^{-2} + 3.1390z^{-3} + 0.9603z^{-4}} \tag{11.97}$$

$$G_4(z) = \frac{0.9270 - 1.2079z^{-1} + 0.9270z^{-2}}{1 - 1.2079z^{-1} + 0.8541z^{-2}} \tag{11.98}$$

Solution:

The MATLAB script to plot each of the transfer functions of (11.95) through (11.98), is given below where $N = 512$, i.e., the default value.

```
b1=[2.8982  8.6946  8.6946  2.8982]*10^(–3); a1=[1  –2.3741  1.9294  –0.5321];...
[G1z,w1T]=freqz(b1,a1);...
%
b2=[0.5276  –1.5828  1.5828  –0.5276]; a2=[1  –1.7600  1.1829  –0.2781];...
[G2z,w2T]=freqz(b2,a2);...
```

```
%
b3=[6.8482  0  −13.6964  0  6.8482]*10^(−4); a3=[1  3.2033  4.5244  3.1390  0.9603];...
[G3z,w3T]=freqz(b3,a3);...
%
b4=[0.9270  −1.2079  0.9270]; a4=[1  −1.2079  0.8541];...
[G4z,w4T]=freqz(b4,a4);...
clf; % clear current figure;...
%
subplot(221), semilogx(w1T,abs(G1z)), axis([0.1 1 0 1]), title('Filter for G1(z)');...
xlabel(''),ylabel('Magnitude'),grid;...
%
subplot(222), semilogx(w2T,abs(G2z)), axis([0.1 10 0 1]), title('Filter for G2(z)');...
xlabel(''),ylabel('Magnitude'),grid;...
%
subplot(223), semilogx(w3T,abs(G3z)), axis([1 10 0 1]), title('Filter for G3(z)');...
xlabel(''),ylabel('Magnitude'),grid;...
%
subplot(224), semilogx(w4T,abs(G4z)), axis([0.1 10 0 1]), title('Filter for G4(z)');...
xlabel(''),ylabel('Magnitude'),grid
```

The plots are shown in Figure 11.42. We observe that the given transfer functions are for low–pass, high–pass, band–pass, and band–elimination digital filters.

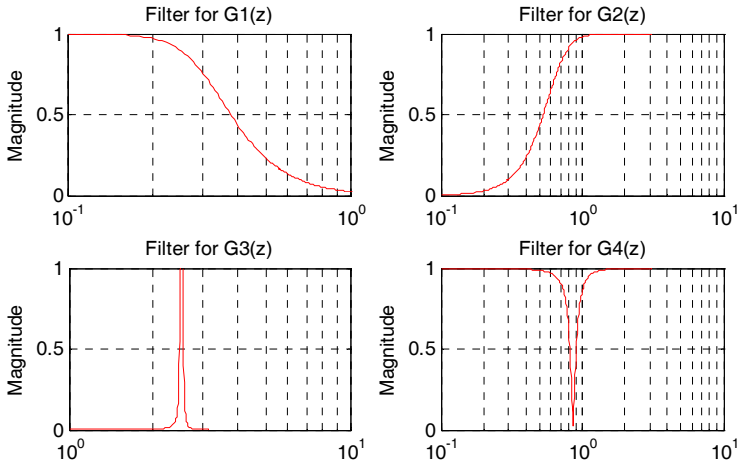


Figure 11.42. Plot for the transfer functions of Example 11.20

Example 11.21

We are given a 165 KHz total bandwidth, and within this bandwidth we must accommodate four different signals. Each of these signals requires 25 KHz bandwidth. We are asked to define the types of filters and cutoff frequencies to avoid interference among these signals.

Solution:

We will use Butterworth filters up to order 12 to obtain sharp cutoffs, and the following types and bandwidths for each.

1. Low–pass filter with bandwidth 0 to 25 KHz, (3 dB cutoff at 25 KHz)

2. Band–pass filter with bandwidth from 40 KHz to 65 KHz

3. High–pass filter with 3 dB frequency at 90 KHz

4. Band–elimination filter with stop-band from 115 KHz to 140 KHz

The highest (Nyquist) frequency is 165 KHz so we choose a sampling frequency of 330 KHz.

The MATLAB **freqz** function in the script below normalizes the frequencies from 0 to π where π = Nyquist frequency.

```
% N=512;          % Default;...
fs=330000;        % Chosen sampling frequency;...
Ts=1/fs;          % Sampling period;...
fn=fs/2;          % Nyquist frequency
%
f1=25000/fn;      % Low–pass filter cutoff frequency (Signal 1 End);...
f2=40000/fn;      % Band–pass left cutoff frequency (Signal 2 Start);...
f3=65000/fn;      % Band–pass right cutoff frequency (Signal 2 End);...
f4=90000/fn;      % High–pass filter cutoff frequency (Signal 3 Start);...
f5=115000/fn;     % Band–stop filter left cutoff frequency (Signal 3 End);...
f6=140000/fn;     % Band–stop filter right cutoff frequency (Signal 4 Start);...
% Signal 4 will terminate at 165 kHz
[b1,a1]=butter(12,f1);...
[b2,a2]=butter(12,[f2,f3]);...
[b3,a3]=butter(12,f4,'high');...
[b4,a4]=butter(12,[f5,f6],'stop');...
%
[G1z,wT]=freqz(b1,a1);...
[G2z,wT]=freqz(b2,a2);...
[G3z,wT]=freqz(b3,a3);...
[G4z,wT]=freqz(b4,a4);...
%
Hz=wT/(2*pi*Ts);...
%
clf; % clear current figure;...
%
plot(Hz,abs(G1z),Hz,abs(G2z),Hz,abs(G3z),Hz,abs(G4z)), axis([0 16*10^4 0 1]);...
title('Four signals separated by four digital filters');...
xlabel('Hz'); ylabel('Magnitude'); grid
```

The plot of Figure 11.43 shows the frequency separations for these four signals.

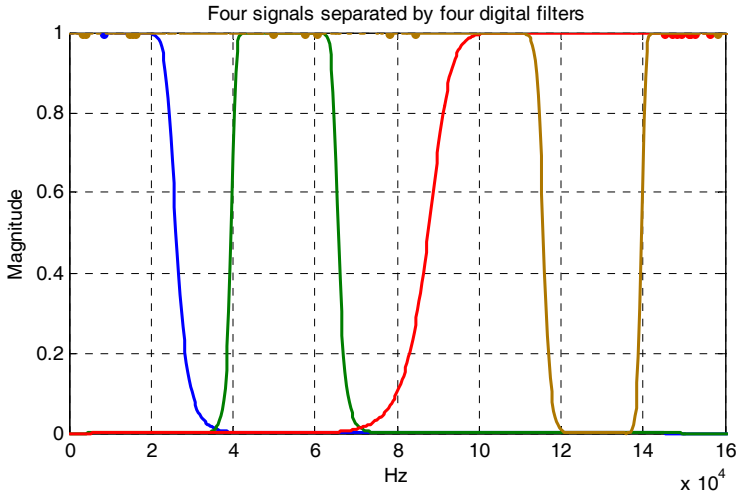


Figure 11.43. Frequency separations for the signals of Example 11.21

In the following example, we will demonstrate the MATLAB **filter** function that is being used to remove unwanted frequency components from a function. But before we use the **filter** function, we must design a filter that is capable of removing those unwanted components.

Example 11.22

In Chapter 7, Subsection 7.4.4, Page 7–20, we found that the half–wave rectifier with no symmetry can be represented by the trigonometric Fourier series

$$f(t) = \frac{A}{\pi} + \frac{A}{2}\sin t - \frac{A}{\pi}\left[\frac{\cos 2t}{3} + \frac{\cos 4t}{15} + \frac{\cos 6t}{35} + \frac{\cos 8t}{63} + \ldots\right]$$

In this example, we want to filter out just the first 2 terms, in other words, to remove all cosine terms. To simplify this expression, we let $A = 3\pi$ and we truncate it by eliminating all cosine terms except the $\cos 2t$ term. Then,

$$g(t) = 3 + 1.5\sin t - \cos 2t \tag{11.99}$$

The problem now reduces to design a low–pass digital filter, and use the **filter** command to remove the cosine term in (11.99).

Solution:

We will use a $6-$pole digital low–pass Butterworth filter because we must have a sharp transition between the 1 and 2 rad/s frequency range. Also, since the highest frequency component is 2 rad/s, to avoid aliasing, we must specify a sampling frequency of $\omega_S = 4$ rad/s. Thus, the

sampling frequency must be $f_S = \omega_S/2\pi = 2/\pi$ and therefore, the sampling period will be $T_S = 1/f_S = \pi/2$. We choose $T_S = 0.5$; this is sufficiently small. Also, we choose the cutoff frequency of the filter to be $\omega_C = 1.5$ rad/s in order to attenuate the cosine terms.

The MATLAB script below will perform the following steps:

1. Will compute coefficients of the numerator and denominator of the transfer function with normalized cutoff frequency.
2. Will recompute the coefficients for the desired frequency.
3. Use the **bilinear** function to map the analog transfer function to a digital transfer function, and will plot the frequency response of the digital filter.
4. Will recompute the digital filter transfer function to account for the warping effect.
5. Will use the **filter** function to remove the cosine terms

```
% Step 1;...
%
[z,p,k]=buttap(6);...
[b,a]=zp2tf(z,p,k);...
%
% Step 2;...
%
wc=1.5;                         % Chosen cutoff frequency;...
[b1,a1]=lp2lp(b,a,wc);          % Convert to actual cutoff frequency;...
%
% Step 3;...
%
T=0.5;                          % Define sampling period;...
[Nz,Dz]=bilinear(b1,a1,1/T);    % Map to digital filter using the bilinear transformation;...
w=0:2*pi/300:pi;                % Define range for plot;...
Gz=freqz(Nz,Dz,w);              % The digital filter transfer function;...
%
clf;...
%
plot(w,abs(Gz)); axis([0 2 0 1]); grid; hold on;...
% We must remember that when z is used as a function of;...
% normalized frequency, the range of frequencies of G(z) are;...
% from zero to pi and the normalized cutoff frequency on the;...
% plot is wc*T=1.5*0.5=0.75 r/s;...
%
xlabel('Radian Frequency w in rads/sec'),...
ylabel('Magnitude of G(z)'),...
title('Digital Filter Response in Normalized Frequency, Example 11.22');...
%
fprintf('Press any key to continue \n');...
```

```
%
pause;...
%
% Step 4;...
%
p=6; T=0.5;                % Number of poles and Sampling period;...
wc=1.5;                    % Analog cutoff frequency in rad/sec;...
wd=wc*T/pi;                % Normalized digital filter cutoff frequency by (11.88), Page 11–55;...
[Nzp,Dzp]=butter(p,wd);...
fprintf('Summary: \n\n');...
fprintf('WITHOUT PREWARPING: \n\n');...
%
fprintf('The num N(z) coefficients in descending order of z are: \n\n');...
fprintf('%8.4f \t',[Nz]);...
fprintf('\n\n');...
fprintf('The den D(z) coefficients in descending order of z are: \n\n');...
fprintf('%8.4f \t',[Dz]);...
fprintf('\n\n');...
fprintf('WITH PREWARPING: \n\n');...
%
fprintf('The num N(z) coefficients in descending order of z are: \n\n');...
fprintf('%8.4f \t',[Nzp]);...
fprintf('\n\n');...
fprintf('The den D(z) coefficients in descending order of z are: \n\n');...
fprintf('%8.4f \t',[Dzp]);...
fprintf('\n\n');...
```

The plot of the low–pass filter that will remove the cosine terms is shown in Figure 11.44.

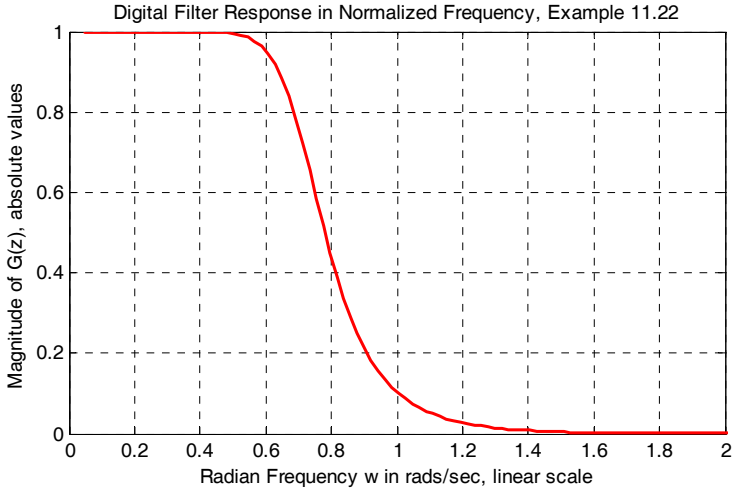


Figure 11.44. Plot of the low-pass filter of Example 11.22

```
Summary:

WITHOUT PREWARPING:

The num N(z) coefficients in descending order of z are:
  0.0007   0.0040   0.0100   0.0134   0.0100   0.0040   0.0007

The den D(z) coefficients in descending order of z are:
  1.0000  -3.2379   4.7566  -3.9273   1.8999  -0.5064   0.0578

WITH PREWARPING:

The num N(z) coefficients in descending order of z are:
  0.0008   0.0050   0.0125   0.0167   0.0125   0.0050   0.0008

The den D(z) coefficients in descending order of z are:
  1.0000  -3.1138   4.4528  -3.5957   1.7075
```

```
% Step 5;...
%
Nzp=[0.0008  0.0050  0.0125  0.0167  0.0125  0.0050  0.0008];...
Dzp=[1.0000 –3.1138  4.4528 –3.5957  1.7075 –0.4479  0.0504];...
n=0:150;...
T=0.5;...
gt=3+1.5*sin(n*T)–cos(2*n*T);...
yt=filter(Nzp,Dzp,gt);...
%
% We will plot the unfiltered analog signal gta;...
%
t=0:0.1:12;...
gta=3+1.5*sin(t)–cos(2*t);...
subplot(211), plot(t,gta), axis([0,12, 0, 6]); hold on;...
xlabel('Continuous Time t'); ylabel('Function g(t)');...
%
% We will plot the filtered analog signal y(t);...
%
subplot(212), plot(n*T,yt), axis([0,12, 0, 6]); hold on;...
xlabel('Continuous Time t'); ylabel('Filtered Output y(t)');...
%
fprintf('Press any key to continue \n'); pause;...
%
% We will plot the unfiltered discrete time signal g(n*T);...
%
subplot(211), stem(n*T,gt), axis([0,12, 0, 6]); hold on;...
xlabel('Discrete Time nT'); ylabel('Discrete Function g(n*T)');...
%
% We will plot the filtered discrete time signal y(n*T);...
subplot(212), stem(n*T,yt), axis([0,12, 0, 6]); hold on;...
xlabel('Discrete Time nT'); ylabel('Filtered Output y(n*T)')
```

The analog and digital inputs and outputs are shown in Figures 11.45 and 11.46 respectively.

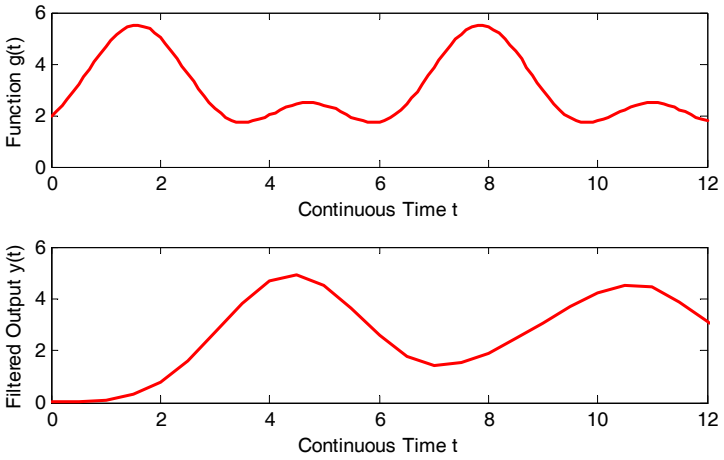


Figure 11.45. Continuous time input and output waveforms for Example 11.22

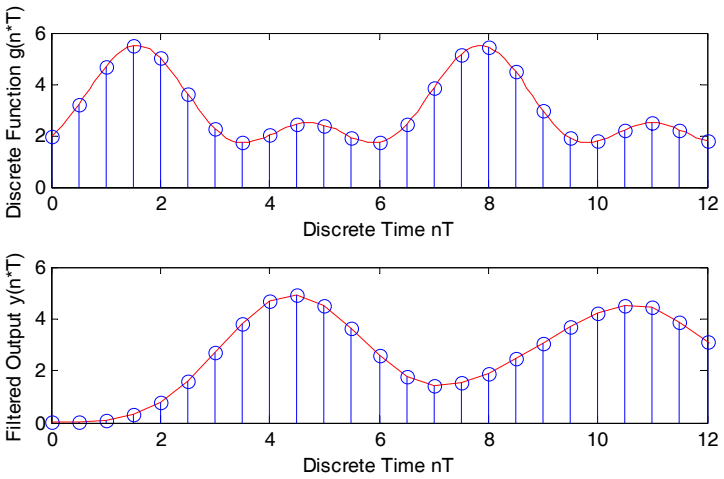


Figure 11.46. Discrete time input and output waveforms for Example 11.22

We conclude this section with one more example to illustrate the use of the MATLAB **find** function. This function displays the subscripts of an array of numbers where a relational expression is true. For example,

```
x=–2:5; % Display the integers in the range -2 <= x < =5
x =
     –2     –1      0      1      2      3      4      5

k=find(x>0.8); % Find the subscripts of the numbers for which x > 0.8
```

```
k =
     4     5     6     7     8
```

y=x(k); % Create a new array y using the indices in k

```
y =
     1     2     3     4     5
```

Example 11.23

Given the function $f(t) = 5\sin 2t - 10\cos 5t$, use the MATLAB **randn** function to add random (Gaussian) noise to $f(t)$ and plot this signal plus noise waveform which we denote as

$$x(t) = f(t) + \text{randn}(N) = 5\sin 2t - 10\cos 5t + \text{randn}(\text{size}(t)) \tag{11.100}$$

where $0 \le t \le 512$. Next, use the **fft** function to compute the frequency components of the 512–point FFT and plot the spectrum of this noisy signal. Finally, use the **find** function to restrict the frequency range of the spectrum to identify the frequency components of the signal $f(t)$.

Solution:

The MATLAB script is shown below.

```
t=linspace(0, 10, 512); x=10*sin(2*t)-5*cos(5*t)+15*randn(size(t));...
% We plot the signal to see what it looks like;...
%
subplot(221); plot(t,x),title('x(t)=Signal plus Noise');...
%
% The input signal x is shown in the upper left corner of the graph;...
%
% Next, we will compute the frequency domain of the signal x;...
%
X=fft(x);...
%
% The sampling period of x is found by the time difference of two samples;...
%
Ts=t(2)-t(1);...
%
% and the sampling frequency is;...
%
Ws=2*pi/Ts;...
%
% As we know, the Nyquist frequency Wn is half the sampling frequency;...
%
Wn=Ws/2;...
%
% Now, we will define the frequency domain axis;...
%
```

```
w=linspace(0,Wn,length(t)/2);...
%
% The magnitude of the positive frequency components Xp are found from:;...
%
Xp=abs(X(1:length(t)/2));...
% We want now to plot Xp versus radian frequency w;...
%
subplot(222); plot(w,Xp),title('Spectrum of Signal & Noise in Wide Range');...
%
% We will select the frequencies of interest with the "find" function:;...
%
k=find(w<=20);...
%
% Now we will plot this restricted range;...
%
subplot(212); plot(w(k), Xp(k)),title('Spectrum of Signal & Noise in Narrow Range');...
%
% The last plot will have grid, labels and title;...
%
xlabel('Frequency, rads/sec'); ylabel('Frequency Components');...
title('Spectrum of Signal & Noise in Narrow Range'); grid
```

The signal is shown in Figure 11.47.

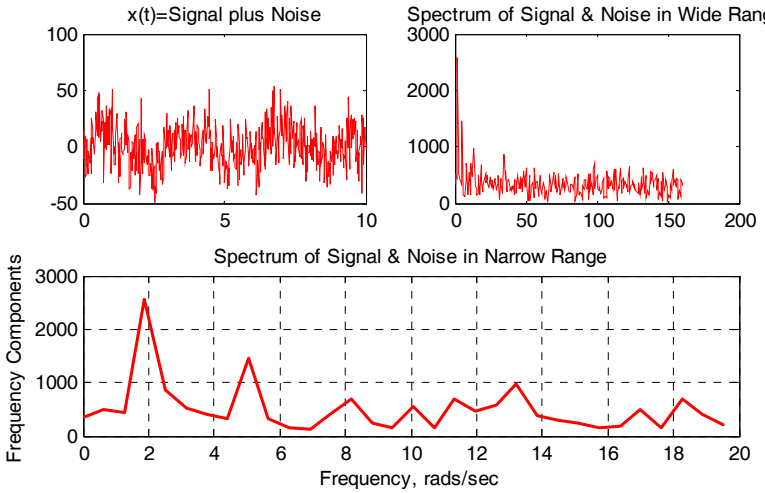


Figure 11.47. Waveforms for Example 11.23

We observe the appearance of the sinusoids at 2 and 5 rad/s in the lower plot. They were undistinguished in the time–domain of the upper left plot. The upper right plot indicates that the signal f(t) has frequency components in the lower range of frequencies, but these cannot be identified precisely.

11.6 Digital Filter Design with Simulink

As stated earlier in this chapter, a digital filter, in general, is a computational process, or algorithm that converts one sequence of numbers representing the input signal into another sequence representing the output signal. Accordingly, a digital filter can perform functions as differentiation, integration, estimation, and, of course, like an analog filter, it can filter out unwanted bands of frequency.

In this section we provide several applications using Simulink models.

A given transfer function H(z) of a digital filter can be realized in several forms, the most common being the **Direct Form I**, **Direct Form II**, **Cascade (Series)**, and **Parallel**. These are described in Subsections 11.6.1 through 11.6.4 below. Subsection 11.6.5 describes the Simulink Digital Filter Design block.

11.6.1 The Direct Form I Realization of a Digital Filter

The **Direct Form I Realization** of a second–order digital filter is shown in Figure 11.48.

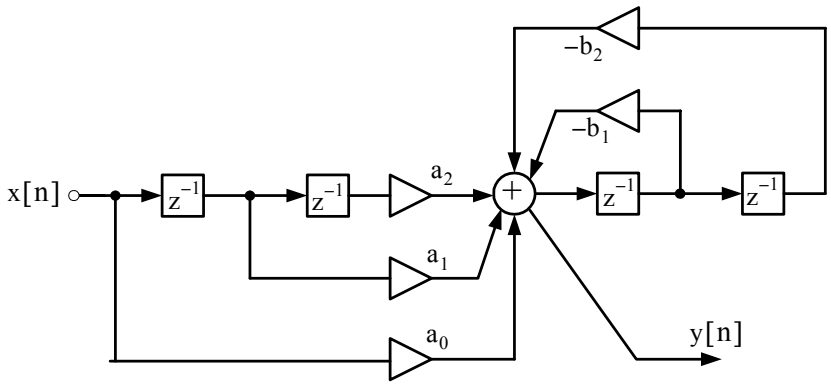


Figure 11.48. Direct Form I Realization of a second–order digital filter

At the summing junction of Figure 11.48 we obtain

$$a_0X(z) + a_1z^{-1}X(z) + a_2z^{-2}X(z) + (-b_1)z^{-1}Y(z) + (-b_2)z^{-1}Y(z) = Y(z)$$

$$X(z)(a_0 + a_1z^{-1} + a_2z^{-2}) = Y(z)(1 + b_1z^{-1} + b_2z^{-2})$$

and thus the transfer function of the Direct Form I Realization of the second-order digital filter of Figure 11.48 is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1z^{-1} + a_2z^{-2}}{1 + b_1z^{-1} + b_2z^{-2}} \tag{11.101}$$

A disadvantage of a Direct Form I Realization digital filter is that it requires 2k registers where k represents the order of the filter. We observe that the second–order (k = 2) digital filter of Fig-

ure 11.48 requires 4 delay (register) elements denoted as z^{-1}. However, this form of realization has the advantage that there is no possibility of internal filter overflow.*

11.6.2 The Direct Form II Realization of a Digital Filter

Figure 11.49 shows the **Direct Form-II**† **Realization** of a second–order digital filter. The Simulink **Transfer Fcn Direct Form II** block implements the transfer function of this filter.

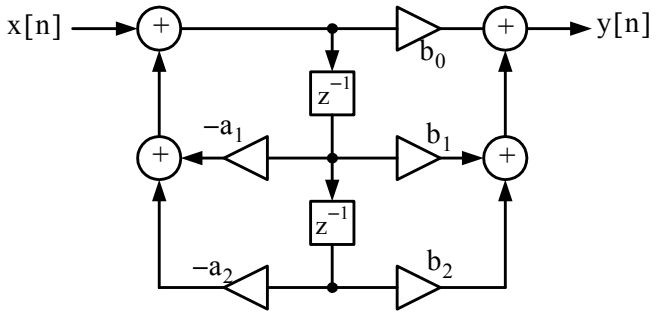


Figure 11.49. Direct Form-II Realization of a second–order digital filter

The transfer function for the Direct Form–II second–order digital filter of Figure 11.49 is the same as for a Direct Form–I second–order digital filter of Figure 11.48, that is,

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{11.102}$$

A comparison of Figures 11.48 and 11.49 shows that whereas a Direct Form–I second–order digital filter is requires $2k$ registers, where k represents the order of the filter, a Direct Form–II second–order digital filter requires only k register elements denoted as z^{-1}. This is because the register (z^{-1}) elements in a Direct Form–II realization are shared between the zeros section and the poles section.

Example 11.24

Figure 11.50 shows a Direct Form–II second–order digital filter whose transfer function is

$$H(z) = \frac{1 + 1.5z^{-1} + 1.02z^{-2}}{1 - 0.25z^{-1} + 0.75z^{-2}} \tag{11.103}$$

* *For a detailed discussion on overflow conditions please refer to Digital Circuit Analysis and Design with an Introduction to CPLDs and FPGAs, ISBN 0–9744239–6–3, Section 10.5, Chapter 10, Page 10–6.*

† *The Direct Form–II is also known as the* **Canonical** *Form.*

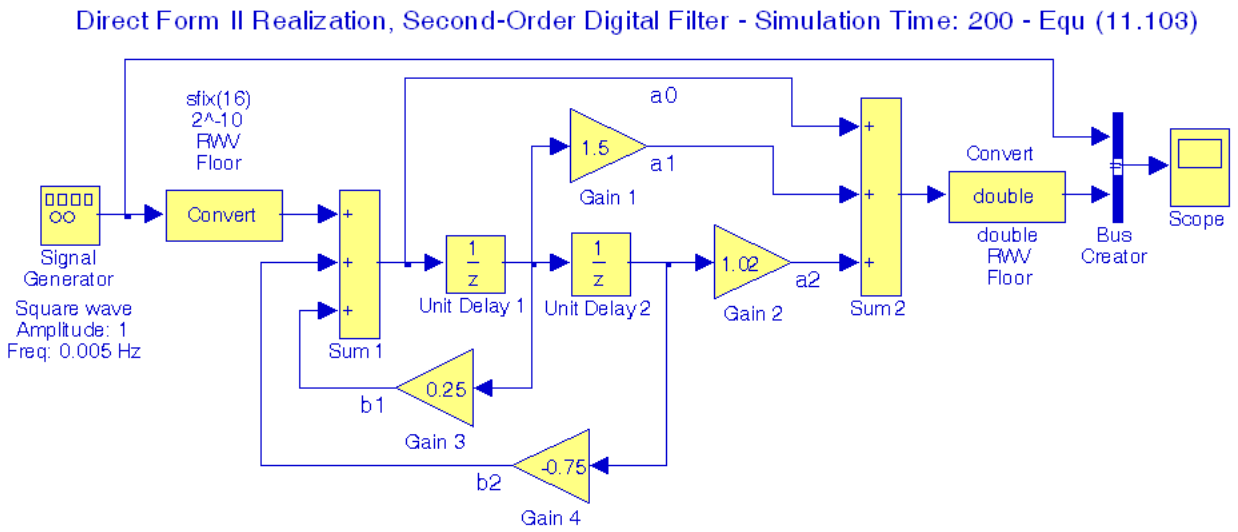


Figure 11.50. Model for Example 11.24

The input and output waveforms are shown in Figure 11.51.

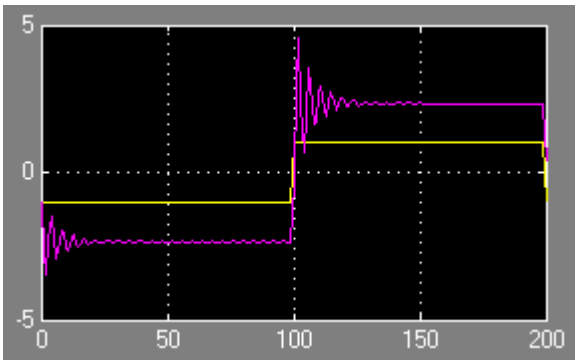


Figure 11.51. Input and output waveforms for the model of Figure 11.50

A demo model using fixed–point Simulink blocks can be displayed by typing

```
fxpdemo_direct_form2
```

in MATLAB's Command prompt. This demo is an implementation of the third–order transfer function

$$H(z) = \frac{1 + 2.2z^{-1} + 1.85z^{-2} + 0.5z^{-3}}{1 - 0.5z^{-1} + 0.84z^{-2} + 0.09z^{-3}}$$

11.6.3 The Series Form Realization of a Digital Filter

For the Series* Form Realization, the transfer function is expressed as a product of first–order and second–order transfer functions as shown in relation (11.104) below.

$$H(z) = H_1(z) \cdot H_2((z) \ldots H_R(z)) \tag{11.104}$$

Relation (11.104) is implemented as the cascaded blocks shown in Figure 11.52.

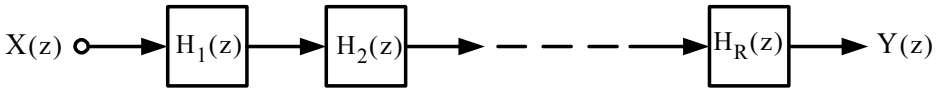


Figure 11.52. Series Form Realization

Figure 11.53 shows the Series Form Realization of a second–order digital filter.

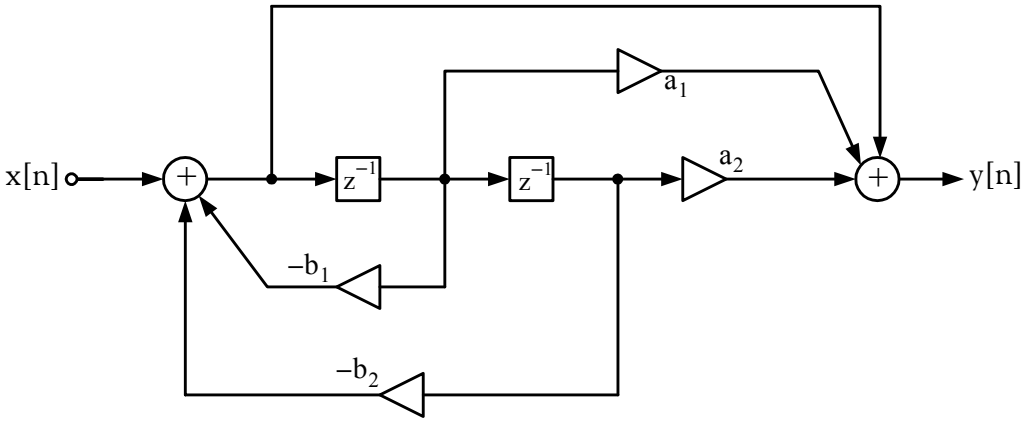


Figure 11.53. Series Form Realization of a second–order digital filter

The transfer function for the Series Form second–order digital filter of Figure 11.53 is

$$H(z) = \frac{1 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{11.105}$$

Example 11.25

The transfer function of the Series Form Realization of a certain second–order digital filter is

$$H(z) = \frac{0.5(1 - 0.36 z^{-2})}{1 + 0.1 z^{-1} - 0.72 z^{-2}}$$

* *The Series Form Realization is also known as the Cascade Form Realization*

To implement this filter, we factor the numerator and denominator polynomials as

$$H(z) = \frac{0.5(1+0.6z^{-1})(1-0.6z^{-1})}{(1+0.9z^{-1})(1-0.8z^{-1})}* \tag{11.106}$$

The model is shown in Figure 11.54, and the input and output waveforms are shown in Figure 11.55.

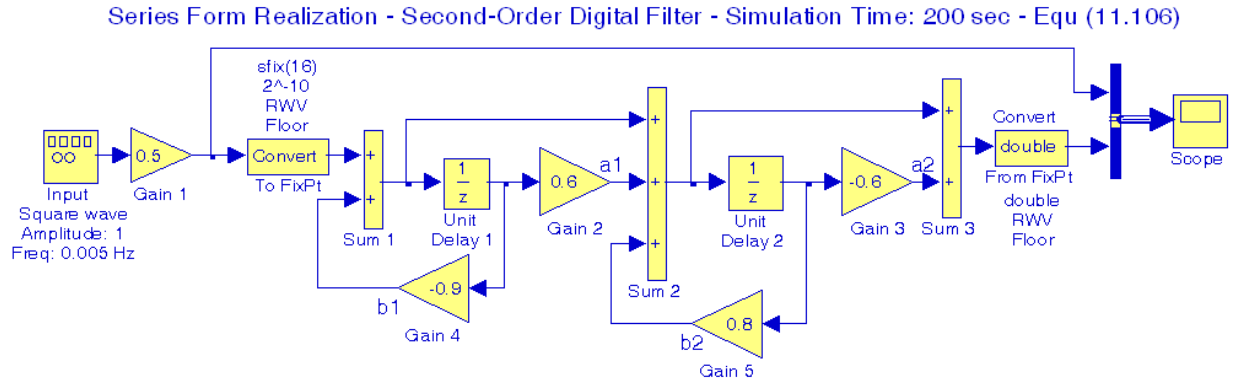


Figure 11.54. Model for Example 11.25

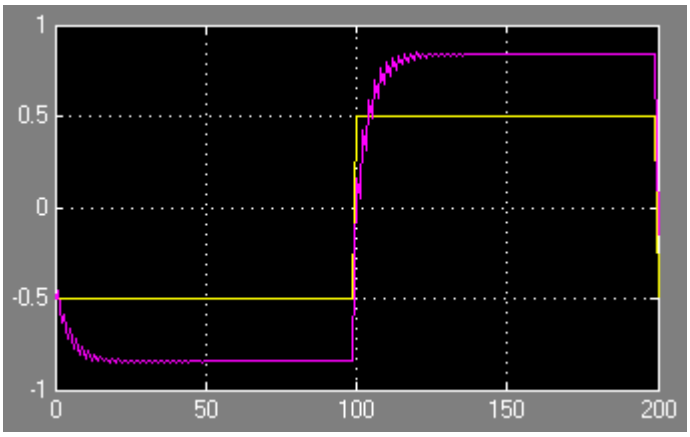


Figure 11.55. Input and output waveforms for the model of Figure 11.54

A demo model using fixed–point Simulink blocks can be displayed by typing

```
fxpdemo_series_cascade_form
```

* *The combination of the of factors in parentheses is immaterial. For instance, we can group the factors as* $\frac{(1+0.6z^{-1})}{(1+0.9z^{-1})}$ *and* $\frac{(1-0.6z^{-1})}{(1-0.8z^{-1})}$ *or as* $\frac{(1+0.6z^{-1})}{(1-0.8z^{-1})}$ *and* $\frac{(1-0.6z^{-1})}{(1+0.9z^{-1})}$

in MATLAB's Command prompt. This demo is an implementation of the third–order transfer function

$$H(z) = \frac{(1 + 0.5z^{-1})(1 + 1.7z^{-1} + z^{-2})}{(1 + 0.1z^{-1})(1 - 0.6z^{-1} + 0.9z^{-2})}$$

11.6.4 The Parallel Form Realization of a Digital Filter

The general form of the transfer function of a Parallel Form Realization is

$$H(z) = K + H_1(z) + H_2(z) + \ldots + H_R(z) \tag{11.107}$$

Relation (11.107) is implemented as the parallel blocks shown in Figure 11.56.

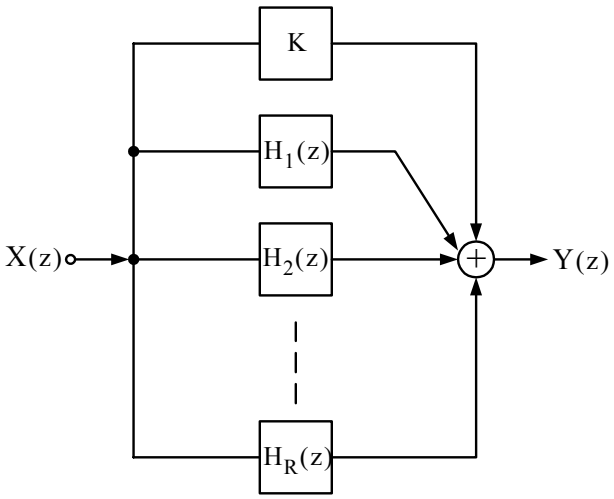


Figure 11.56. Parallel Form Realization

As with the Series Form Realization, the ordering of the individual filters in Figure 11.56 is immaterial. But because of the presence of the constant K, we can simplify the transfer function expression by performing partial fraction expansion after we express the transfer function in the form H(z)/z.

Figure 11.57 shows the Parallel Form Realization of a second–order digital filter. The transfer function for the Parallel Form second–order digital filter of Figure 11.57 is

$$H(z) = \frac{a_1 + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{11.108}$$

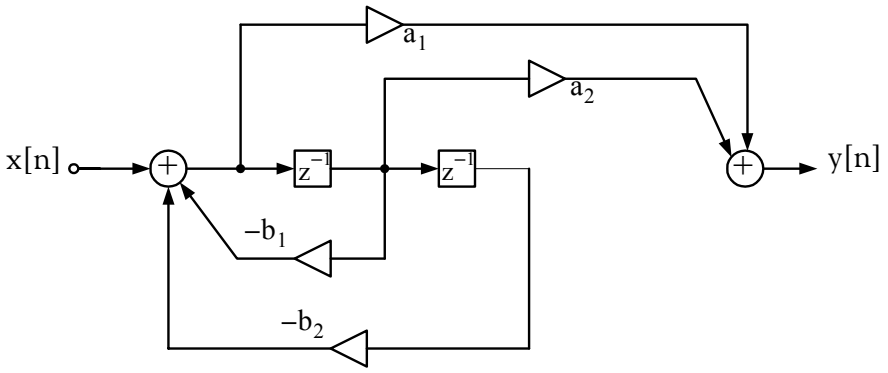


Figure 11.57. Parallel Form Realization of a second–order digital filter

Example 11.26

The transfer function of the Parallel Form Realization of a certain second–order digital filter is

$$H(z) = \frac{0.5(1-0.36z^{-2})}{1+0.1z^{-1}-0.72z^{-2}}$$

To implement this filter, we first express the transfer function as

$$\frac{H(z)}{z} = \frac{0.5(z+0.6)(z-0.6)}{z(z+0.9)(z-0.8)}$$

Next, we perform partial fraction expansion.

$$\frac{0.5(z+0.6)(z-0.6)}{z(z+0.9)(z-0.8)} = \frac{r_1}{z} + \frac{r_2}{(z+0.9)} + \frac{r_3}{(z-0.8)}$$

$$r_1 = \left.\frac{0.5(z+0.6)(z-0.6)}{(z+0.9)(z-0.8)}\right|_{z=0} = 0.25$$

$$r_2 = \left.\frac{0.5(z+0.6)(z-0.6)}{z(z-0.8)}\right|_{z=-0.9} = 0.147$$

$$r_3 = \left.\frac{0.5(z+0.6)(z-0.6)}{z(z+0.9)}\right|_{z=0.8} = 0.103$$

Therefore,

$$\frac{H(z)}{z} = \frac{0.25}{z} + \frac{0.147}{z+0.9} + \frac{0.103}{z-0.8}$$

$$H(z) = 0.25 + \frac{0.147z}{z+0.9} + \frac{0.103z}{z-0.8}$$

$$H(z) = 0.25 + \frac{0.147}{1 + 0.9z^{-1}} + \frac{0.103}{z - 0.8z^{-1}} \tag{11.109}$$

The model is shown in Figure 11.58, and the input and output waveforms are shown in Figure 11.59.

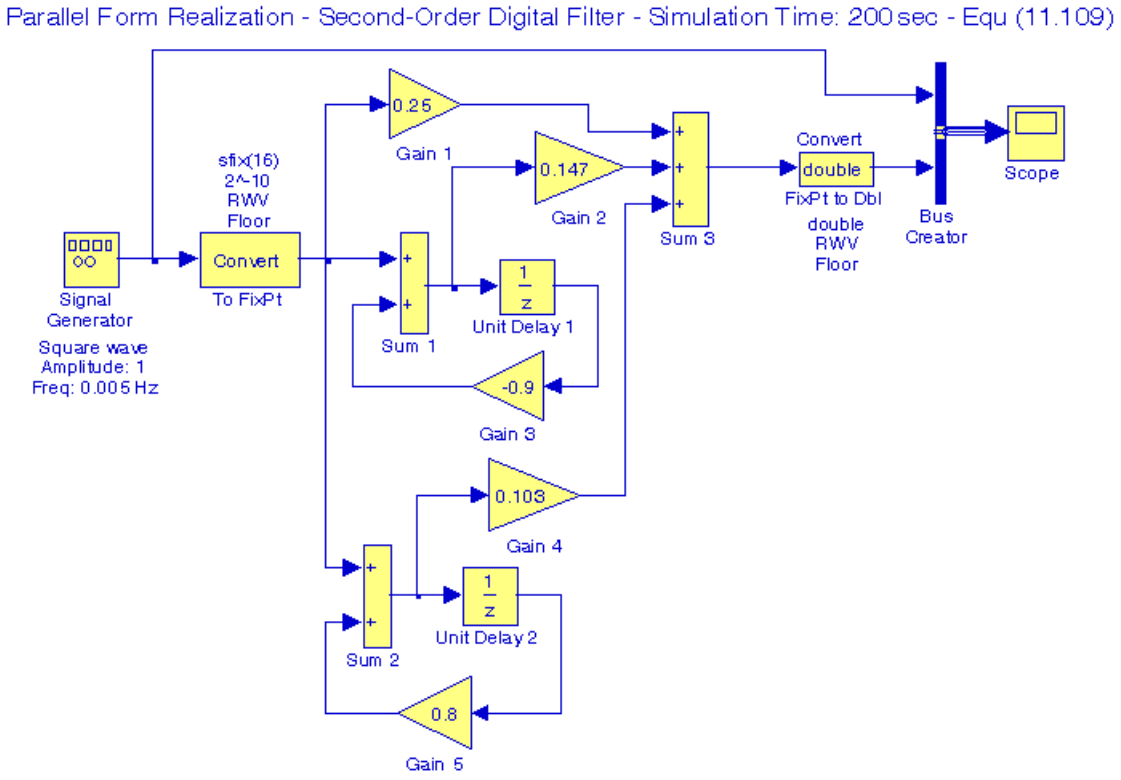


Figure 11.58. Model for Example 11.26

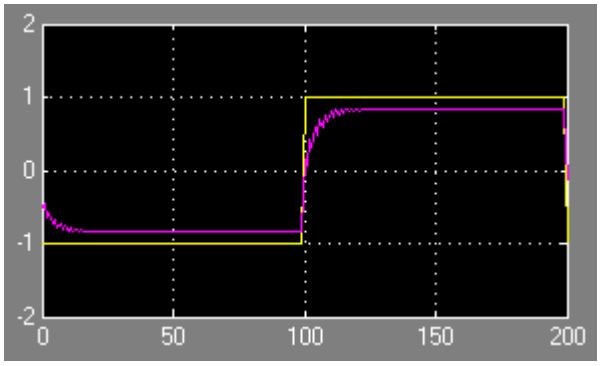


Figure 11.59. Input and output waveforms for the model of Figure 11.58

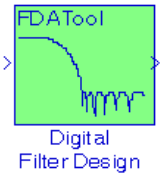
A demo model using fixed–point Simulink blocks can be displayed by typing

fxpdemo_parallel_form

in MATLAB's Command prompt. This demo is an implementation of the third–order transfer function

$$H(z) = 5.5556 - \frac{3.4639}{(1 + 0.1z^{-1})} + \frac{-1.0916 + 3.0086z^{-1}}{1 - 0.6z^{-1} + 0.9z^{-2}}$$

11.6.5 The Digital Filter Design Block



The **Digital Filter Design** block is included in the Simulink Signal Processing Blockset* and requires the installation of the Simulink program to create models related to digital filter design applications. The functionality of this block can be observed by dragging this block into a model and double-clicking it. When this is done, the **Block Parameters** dialog box appears as shown in Figure 11.60. As indicated on the left lower part of this box, we can choose the **Response Type** (Low–Pass, High–Pass, Band–Pass, or Band–Elimination), the **Design Method** (IIR or FIR) where an IIR filter can be Butterworth, Chebyshev Type I, Chebyshev Type II, or Elliptic, and FIR can be Window, Maximally Flat, etc., and the **Window**† can be Kaiser, Hamming, etc. We must click on the Design Filter button at the bottom of the Block Parameters dialog box to update the specifications. Example 11.27 below is very similar to that of Example 11.23, Page 11–68.

Example 11.27

The signal represented by the waveform of Figure 11.61 is the summation of the sinusoidal signals x, y, and z defined in the MATLAB script below.

```
t=0:pi/32:16*pi; x=sin(0.25.*t); y=2.*sin(0.75.*t+pi/6); z=5.*sin(1.5.*t+pi/3); plot(t,x+y+z); grid
```

During transmission of this signal from its source to its destination, this signal is corrupted by the addition of unwanted Gaussian random noise. In this example, we will create a Simulink model that includes a digital filter to remove the Gaussian random noise.

* ***Blocksets*** *are built-in blocks in Simulink that provide a comprehensive block library for different system components. FDA stands for Filter Design and Analysis. This blockset can be obtained from The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, Phone: 508-647-7000, www.mathworks.com.*

† *A window function multiplies the infinite length impulse response (IIR) by a finite width function, referred to as window function, so that the infinite length series will be terminated after a finite number of terms in the series. The most common window functions are described in the Signal Processing Toolbox User's Guide The MathWorks, Inc.*

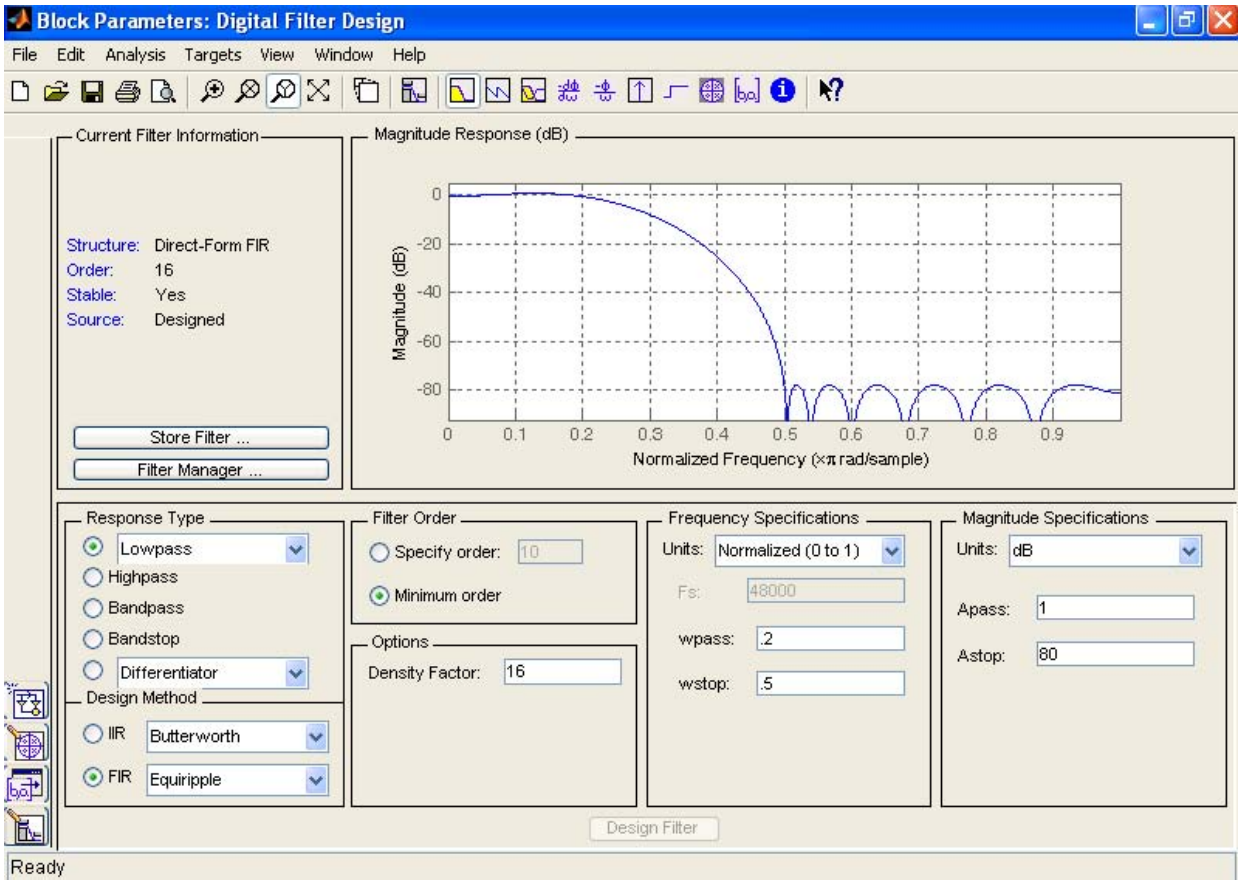


Figure 11.60. The Digital Filter Design Block Parameters dialog box

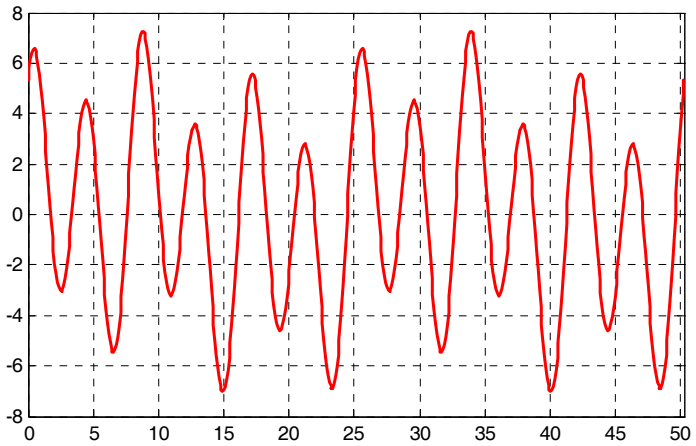


Figure 11.61. Signal to be transmitted for Example 11.27

The Simulink model of Figure11.62 below contains a **Random Source** block to incorporate noise into the system.This block and the three **DSP Sine Wave** blocks are dragged from the **Signal Processing Sources** sub–library under **Signal Management** in the **Signal Processing Blockset** into the model. The **Add** (Sum) and **Scope** blocks are dragged from the **Commonly Used Blocks** main Simulink Library, and the Scope 1 and Scope 2 blocks are configured* with four and two inputs (axes) respectively from the **Scope Parameters** dialog box.

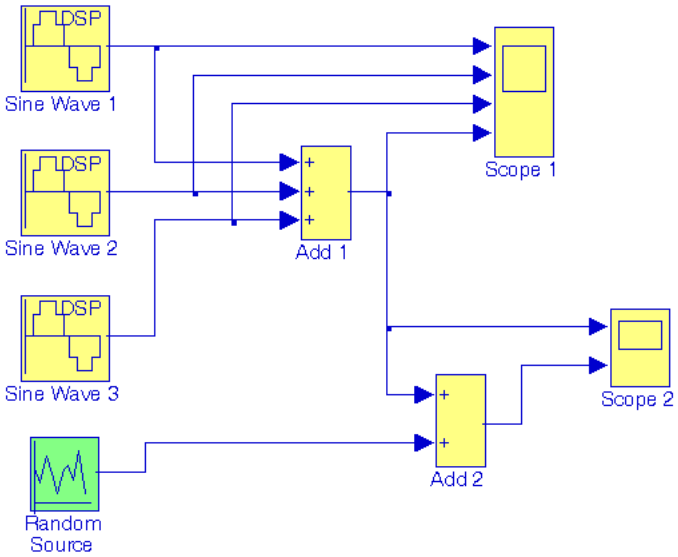


Figure 11.62. Simulink model for Example 11.27

The **DSP Sine Wave** blocks are configured as follows:

DSP Sine Wave 1:

Amplitude: 1, **Freq (Hz)**: 0.25, **Phase**: 0, **Sample Time**: 0.05, All other parameters in their default state

DSP Sine Wave 2:

Amplitude: 2, **Freq (Hz)**: 0.75, **Phase**: pi/6, **Sample Time**: 0.05, All other parameters in their default state

DSP Sine Wave 3:

Amplitude: 5, **Freq (Hz)**: 1.5, **Phase**: pi/3, **Sample Time**: 0.05, All other parameters in their default state

The **Random Source** block is configured as follows:

Source type: Gaussian, **Method**: Ziggurat, **Initial seed**: [23341], **Sample Time**: 0.05

When the simulation command is executed, the **Scope 1** block in the model of Figure 11.62, dis-

* *For a detailed discussion on configuration of the Scope block, please refer to Introduction to Simulink with Engineering Applications, ISBN 0–9744239–7–1.*

plays the input and output waveforms shown in Figure 11.63, and the **Scope 2** block displays the input and output waveforms shown in Figure 11.64.

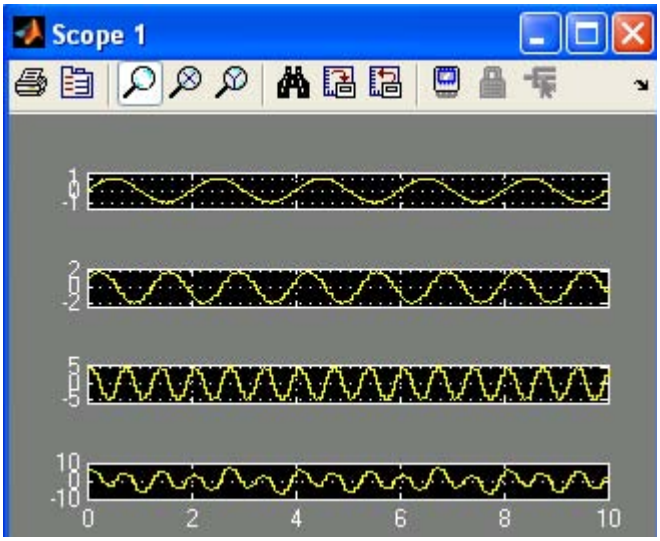


Figure 11.63. Input and output waveforms displayed in Scope 1 block of Figure 11.62

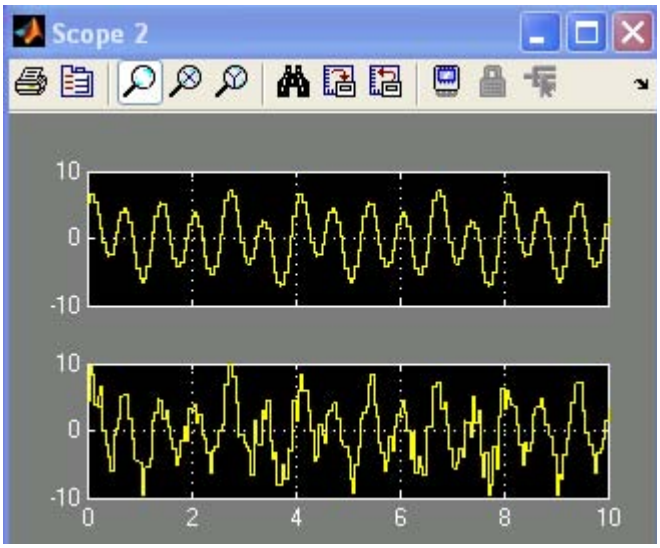


Figure 11.64. Input and output waveforms displayed in Scope 2 block of Figure 11.62

Next, we add an **FDA Tool Digital Filter Design** block as shown in Figure 11.65 to remove the unwanted noise created by the Random Source block. The Block Parameters dialog box for the **FDA Tool Digital Filter Design** block is configured as follows:

Response Type: Lowpass, **Design Method**: FIR, Window, **Window**: Rectangular, and all other unlisted parameters in their default state. Of course, we can choose any other design options. With those specifications, the Scope 2 block displays the waveforms shown in Figure

11.66.

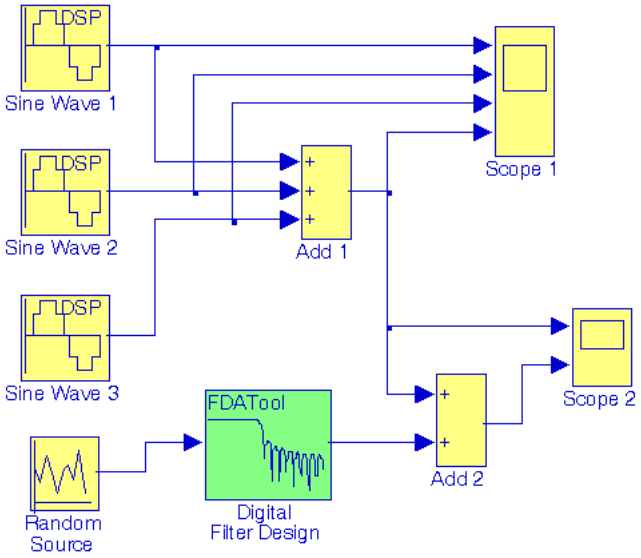


Figure 11.65. The model of Figure 11.62 with the addition of an FDA Tool Digital Filter Design block.

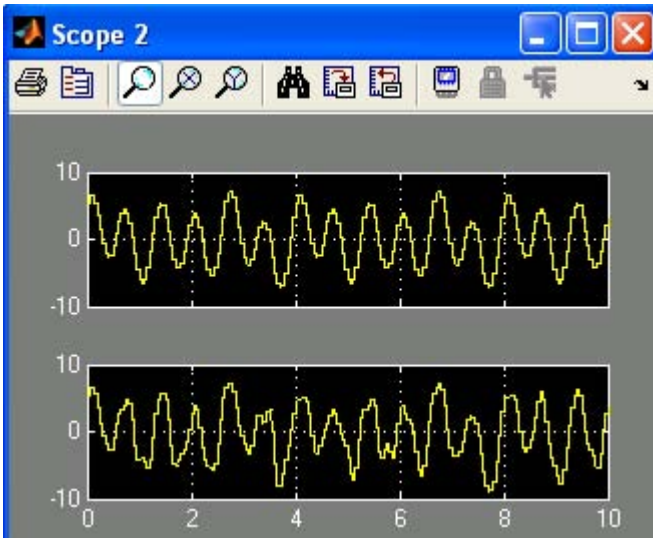


Figure 11.66. Input and output waveforms displayed in Scope 2 block of Figure 11.65

Figure 11.66 reveals that with the addition of the **FDA Tool Digital Filter Design** block, most of the unwanted noise has been removed. However, we can remove the remaining noise with the addition of an *adaptive filter*.* The Signal Processing Blockset contains blocks that implement the Least–Mean–Square (LMS) block, the Fast Block LMS, and Recursive Least Squares (RLS)

* *An **adaptive filter** is a digital filter that performs digital signal processing and can adapt its performance based on the input signal. All filters we've considered thus far are non–adaptive filter and their characteristics are defined by their transfer function.*

adaptive filter algorithms. For our example, we will add an LMS adaptive filter to the model of Figure 11.65. Thus, from the Signal Processing Blockset, we click on the Filtering Library, then on the Adaptive Filters sub–library, and we drag the **LMS Filter** block into our model which is connected as shown in Figure 11.67 where the **Wts** (Weights) port of the **LMS Filter** block which outputs the filter weights is left unconnected. The waveforms displayed by the Scope 3 block are shown in Figure 11.68 where last waveform indicates the output of the Error port which is the difference between the desired signal of the LMS filter and its output.

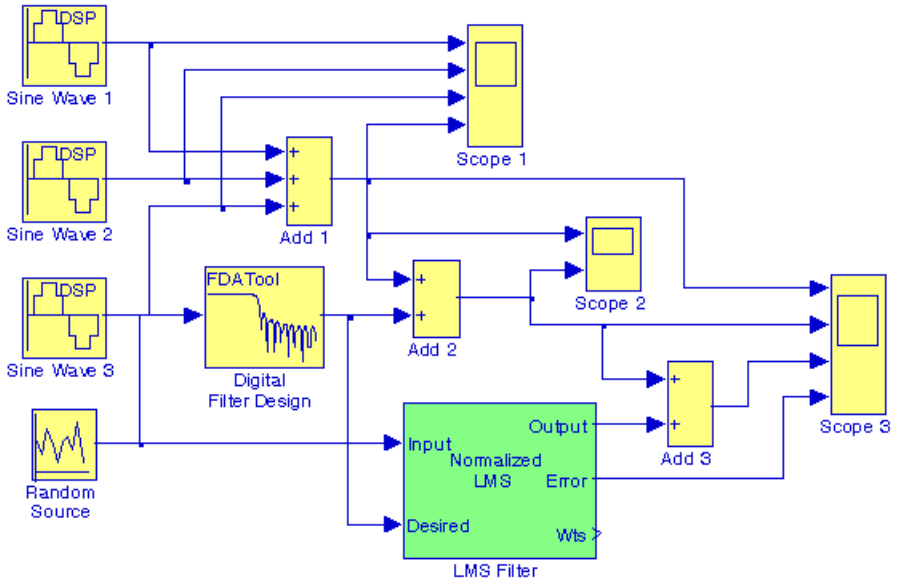


Figure 11.67. Model for Example 11.27 with LMS Filter block

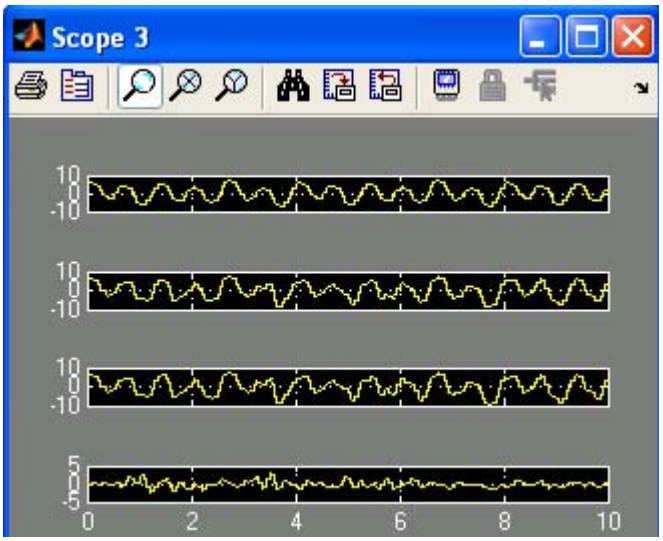


Figure 11.68. Waveforms displayed by Scope 3 block in the model of Figure 11.67

The error is never exactly zero and thus the adaptive filter continuously modifies the filter coefficients to provide a better approximation of the noise. We can view these coefficients as they change with time by connecting a **Vector Scope** block to the **Wts** output port of the **LMS filter** block as shown in Figure 11.69.

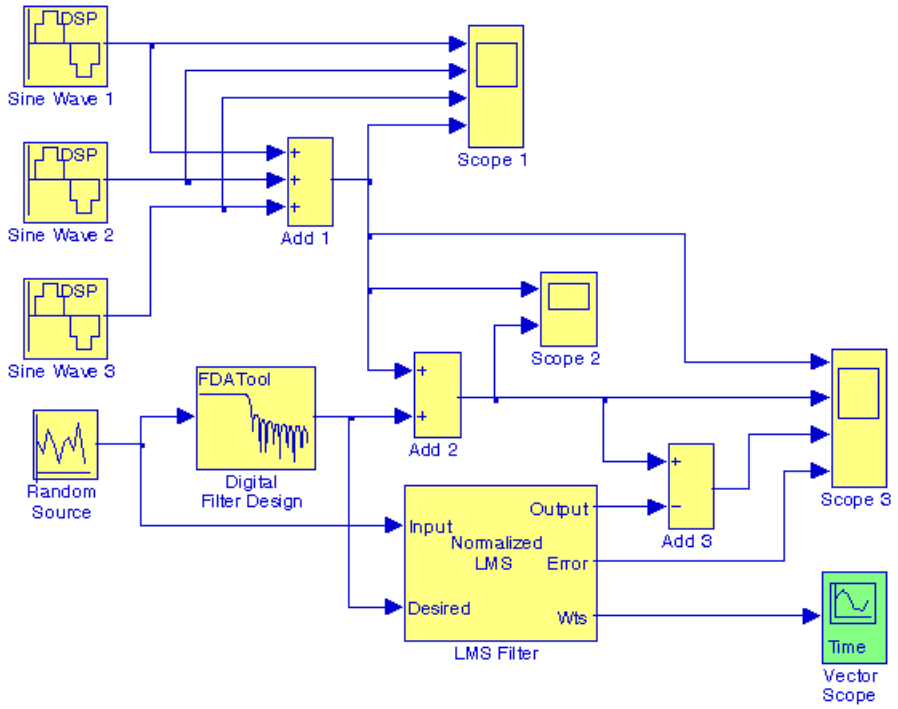


Figure 11.69. Model for Example 11.27 with Vector Scope block

The Block Parameters dialog box for the **Vector Scope** block contains four tabs, and for the model of Figure 11.69 these are configured as follows:

Scope Properties tab: Input domain: **Time**, Time display span (number of frames): **1**

Display Properties tab: Select the following check boxes: **Show grid, Frame Number, Compact Display,** and **Open scope at start of simulation**

Axis Properties tab: Minimum Y–limit: **–0.1**, Maximum Y–limit: **0.5**, Y–axis title: **Filter Weights**

Line Properties tab: Line visibilities: **on**, Line style: ––, Line markers: **o**, Line colors: **[1 0 0]**

Before execution of the simulation command, the configuration parameters are specified as follows:

From the **Simulation** drop menu, in the **Solver** pane, for the **Stop time** parameter, we enter **inf**. From the **Type** list, we choose **Fixed–step**, and from the **Solver** list we choose **discrete (no continuous states)**. We close the configuration parameters dialog box by clicking **OK**.

When the simulation command is issued, we observe that the **Vector Scope** window opens automatically, and the filter coefficients change with time and eventually approach their steady–state

values as shown in Figure 11.70.

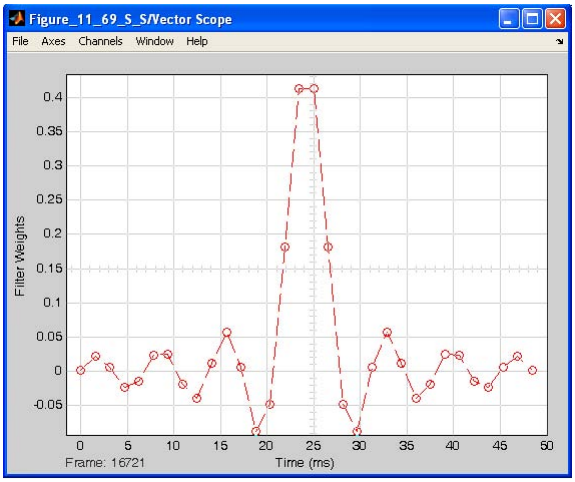


Figure 11.70. The filter coefficients displayed in the Vector Scope Window in the model of Figure 11.69

Next, we double–click on the **Scope 3** block in the model of Figure 11.69, and after some time we observe that the error decreases to zero as shown in Figure 11.71, and the output of the adaptive LMS filter is practically the same as the original input signal.

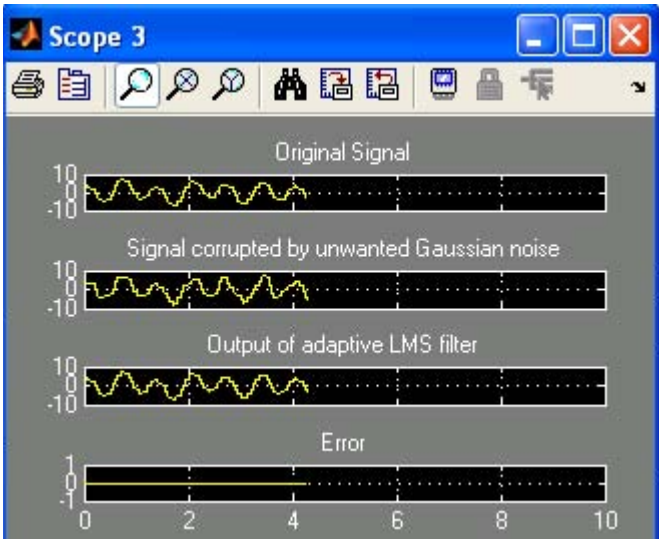


Figure 11.71. Waveforms displayed in the Scope 3 in the model of Figure 11.69

Since for the **Stop time** parameter we have specified **inf**, the simulation time goes on forever, and we observe that the waveforms in Figure 11.71 are updated continuously. To stop the simulation, we must click on the **Stop simulation** icon which is indicated by a small black square immediately to the left of the **Stop time** field in the window of our model. The **Stop simulation** is active only when the **Stop time** is specified as **inf** and the simulation command has been issued. To

pause the simulation, we must click on the **Pause simulation** icon indicated by two small vertical bars immediately to the left of the **Stop simulation** icon.

11.7 Summary

- Analog filters are defined over a continuous range of frequencies. They are classified as low–pass, high–pass, band–pass and band–elimination (stop–band).
- An all–pass or phase shift filter has a constant magnitude response but is phase varies with frequency.
- A digital filter, in general, is a computational process, or algorithm that converts one sequence of numbers representing the input signal into another sequence representing the output signal.
- A digital filter, besides filtering out unwanted bands of frequency, can perform functions of differentiation, integration, and estimation.
- Analog filter functions have been used extensively as prototype models for designing digital filters.
- An analog filter can also be classified as passive or active. Passive filters consist of passive devices such as resistors, capacitors and inductors. Active filters are, generally, operational amplifiers with resistors and capacitors connected to them externally.
- If two frequencies ω_1 and ω_2 are such that $\omega_2 = 2\omega_1$, we say that these frequencies are separated by one octave, and if $\omega_2 = 10\omega_1$, they are separated by one decade.
- The analog low–pass filter is used as a basis. Using transformations, we can derive high–pass and the other types of filters from a basic low–pass filter.
- In this chapter we discussed the Butterworth, Chebyshev Type I & II, and Cauer (elliptic) filters.
- The first step in the design of an analog low–pass filter is to derive a suitable *magnitude–squared function* $A^2(\omega)$, and from it derive a $G(s)$ function such that

$$A^2(\omega) = G(s) \cdot G(-s)\big|_{s = j\omega}$$

- The general form of the magnitude–square function $A^2(\omega)$ is

$$A^2(\omega) = \frac{C(b_k\omega^{2k} + b_{k-1}\omega^{2k-2} + \ldots + b_0)}{a_k\omega^{2k} + a_{k-1}\omega^{2k-2} + \ldots + a_0}$$

  where C is the DC gain, a and b are constant coefficients, and k is a positive integer denoting the order of the filter.
- The magnitude–squared function of a Butterworth analog low–pass filter is

$$A^2(\omega) = \frac{1}{(\omega/\omega_C)^{2k} + 1}$$

where k is a positive integer indicating the order of the filter, and ω_C is the cutoff (3 dB) frequency.

- All Butterworth filters have the property that all poles of the transfer functions that describes them, lie on a circumference of a circle of radius ω_C, and they are $2\pi/2k$ radians apart. Thus, if $k = \text{odd}$, the poles start at zero radians, and if $k = \text{even}$, they start at $2\pi/2k$. But regardless whether k is odd or even, the poles are distributed in symmetry with respect to the $j\omega$ axis. For stability, we choose the poles of the left half of the s–plane to form $G(s)$.
- The general form of any analog low–pass (Butterworth, Chebyshev, Elliptic, etc.) filter is

$$G(s)\big|_{lp} = \frac{b_0}{a_k s^k + \ldots + a_2 s^2 + a_1 s + a_0}$$

- The MATLAB **buttap** and **zp2tf** functions are very useful functions in the design of Butterworth filters. The first returns the zeros, poles, and gain for an $N-th$ order normalized prototype Butterworth analog low–pass filter. The resulting filter has N poles around the unit circle in the left half plane, and no zeros. The second performs the zero–pole to transfer function conversion.
- The Chebyshev Type I filters are based on approximations derived from the Chebyshev polynomials $C_k(x)$ that constitute a set of orthogonal functions. The coefficients of these polynomials are tabulated in math tables.
- We can use the MATLAB **cheb1ap** function to design a Chebyshev Type I analog low–pass filter. Thus, the **[z,p,k] = cheb1ap(N,Rp)** statement where **N** denotes the order of the filter, returns the zeros, poles, and gain of an $N-th$ order normalized prototype Chebyshev Type I analog low–pass filter with ripple **Rp** decibels in the pass band.
- The Chebyshev Type II, also known as Inverted Chebyshev filter, is characterized by the following magnitude–square approximation

$$A^2(\omega) = \frac{\varepsilon^2 C_k^2(\omega_C/\omega)}{1 + \varepsilon^2 C_k^2(\omega_C/\omega)}$$

and has the ripple in the stop–band as opposed to Chebyshev Type I which has the ripple in the pass–band. The frequency ω_C defines the beginning of the stop band.

- The elliptic (Cauer) filters are characterized by the low–pass magnitude–squared function

$$A^2(\omega) = \frac{1}{1 + R_k^2(\omega/\omega_C)}$$

where $R_k(x)$ represents a rational elliptic function used with elliptic integrals. Elliptic filters have ripple in both the pass–band and the stop–band.

- We can design elliptic low–pass filters with the MATLAB **ellip** function. The statement **[b,a] = ellip(N,Rp,Rs,Wn,'s')** where **N** is the order of the filter, designs an $N-th$ order low–pass filter with ripple **Rp** decibels in the pass band, a stop band with ripple **Rs** decibels, **Wn** is the cutoff frequency, and **'s'** is used to specify analog elliptic filters. If **'s'** is not included in the above statement, MATLAB designs a digital filter.
- Transformation methods have been developed where a low–pass filter can be converted to another type of filter simply by transforming the complex variable s. These transformations are listed in Table 11.5 where ω_C is the cutoff frequency of a low–pass filter.
- We can use the MATLAB **lp2lp**, **lp2hp**, **lp2bp**, and **lp2bs** functions to transform a low–pass filter with normalized cutoff frequency, to another low–pass filter with any other specified frequency, or to a high–pass filter, or to a band–pass filter, or to a band–elimination filter respectively
- We can use the MATLAB function **bode(num,den)** to generate both the magnitude and phase responses of any transfer function describing the filter type.
- Digital filters are classified in terms of the duration of the impulse response, and in forms of realization.
- An Infinite Impulse Response (IIR) digital filter has infinite number of samples in its impulse response $h[n]$
- A Finite Impulse Response (FIR) digital filter has a finite number of samples in its impulse response $h[n]$
- In a Recursive Realization digital filter the output is dependent on the input and the previous values of the output. In a recursive digital filter, both the coefficients a_i and b_i are present.
- In a Non–Recursive Realization digital filter the output depends on present and past values of the input only. In a non–recursive digital filter, only the coefficients a_i are present, that is, $b_i = 0$.
- Generally, IIR filters are implemented by recursive realization, whereas FIR filters are implemented by non–recursive realization.
- Transformation methods are also available to map an analog prototype to an equivalent digital filter. Three well known methods are the following:

1. The Impulse Invariant Method that produces a digital filter whose impulse response consists of the sampled values of the impulse response of an analog filter.
2. The Step Invariant Method that produces a digital filter whose step response consists of the sampled values of the step response of an analog filter.
3. The Bilinear Transformation that uses the transformation

$$s = \frac{2}{T_s} \cdot \frac{z-1}{z+1}$$

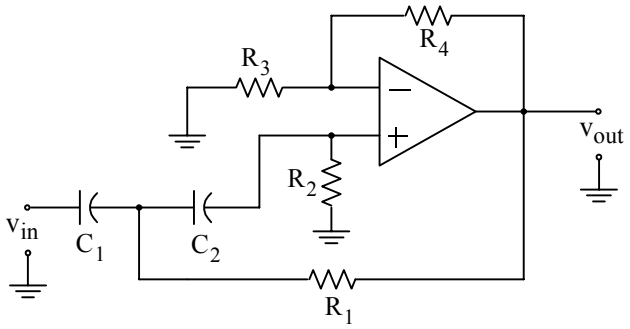
- The analog frequency to digital frequency transformation results in a non–linear mapping; this condition is known as warping.
- The effect of warping can be eliminated by pre–warping the analog filter prior to application of the bilinear transformation.
- We can use the MATLAB **freqz(b,a,N)** function to plot the magnitude of G(z)
- An analog filter transfer function can be mapped to a digital filter transfer function directly with the MATLAB **bilinear(b,a,Fs)** function.
- The MATLAB **filter(b,a,X)** function can be used to remove unwanted frequency components from a function.
- We can use the MATLAB **find(X)** function to restrict the frequency range of the spectrum in order to identify the frequency components of the signal f(t).
- The **Digital Filter Design** block is included in the Simulink Signal Processing Blockset and requires the installation of the Simulink program to create models related to digital filter design applications.

11.8 Exercises

1. The op amp circuit below is a VCVS second–order high–pass filter whose transfer function is

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{Ks^2}{s^2 + (a/b)\omega_C s + (1/b)\omega_C^2}$$

and for given values of a, b, and desired cutoff frequency ω_C, we can calculate the values of C_1, C_2, R_1, R_2, R_3, and R_4 to achieve the desired cutoff frequency ω_C.



For this circuit,

$$R_2 = \frac{4b}{C_1\left\{a + [\sqrt{a^2 + 8b(K-1)}]\right\}\omega_C}$$

$$R_1 = \frac{b}{C_1^2 R_2 \omega_C^2}$$

$$R_3 = \frac{KR_2}{K-1}, \quad K \neq 1$$

$$R_4 = KR_2$$

and the gain K is

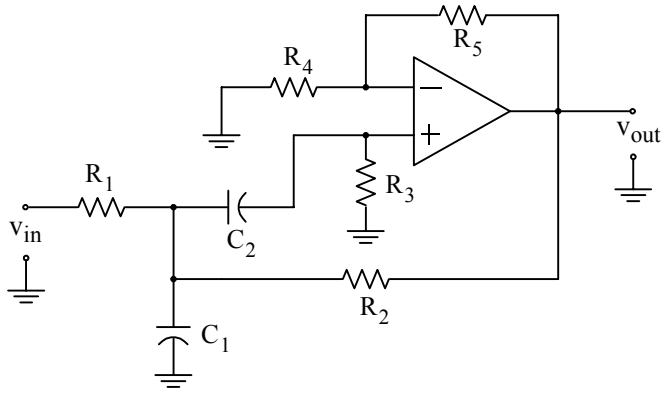
$$K = 1 + R_4/R_3$$

Using these relations, compute the appropriate values of the resistors to achieve the cutoff frequency $f_c = 1$ KHz. Choose the capacitors as $C_1 = 10/f_c$ μF and $C_2 = C_1$. Plot $|G(s)|$ versus frequency.

Solution using MATLAB is highly recommended.

2. The op amp circuit below is a VCVS second–order band–pass filter whose transfer function is

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{K[BW]s}{s^2 + [BW]s + \omega_0^2}$$



Let ω_0 = center frequency, ω_2 = upper cutoff frequency, ω_1 = lower cutoff frequency, Bandwidth BW = $\omega_2 - \omega_1$, and Quality Factor Q = ω_0/BW

We can calculate the values of C_1, C_2, R_1, R_2, R_3, and R_4 to achieve the desired centered frequency ω_0 and bandwidth BW. For this circuit,

$$R_1 = \frac{2Q}{C_1\omega_0 K}$$

$$R_2 = \frac{2Q}{C_1\omega_0\left\{-1 + \sqrt{(K-1)^2 + 8Q^2}\right\}}$$

$$R_3 = \frac{1}{C_1^2\omega_0^2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right)$$

$$R_4 = R_5 = 2R_3$$

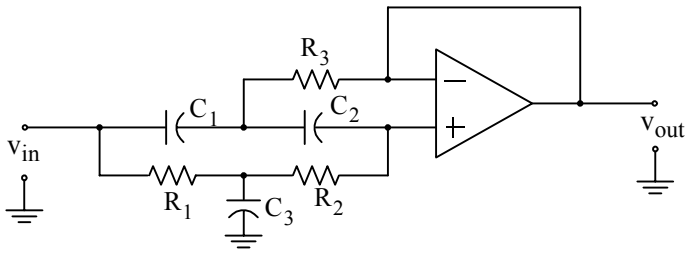
Using these relations, compute the appropriate values of the resistors to achieve center frequency f_0 = 1 KHz, Gain K = 10, and Q = 10.

Choose the capacitors as $C_1 = C_2 = 0.1$ μF. Plot |G(s)| versus frequency.

Solution using MATLAB is highly recommended.

3. The op amp circuit below is a VCVS second–order band–elimination filter whose transfer function is

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{K(s^2 + \omega_0^2)}{s^2 + [BW]s + \omega_0^2}$$



Let ω_0 = center frequency, ω_2 = upper cutoff frequency, ω_1 = lower cutoff frequency, Bandwidth BW = $\omega_2 - \omega_1$, Quality Factor Q = ω_0/BW, and gain K = 1

We can calculate the values of C_1, C_2, R_1, R_2, R_3, and R_4 to achieve the desired centered frequency ω_0 and bandwidth BW. For this circuit,

$$R_1 = \frac{1}{2\omega_0 Q C_1}$$

$$R_2 = \frac{2Q}{\omega_0 C_1}$$

$$R_3 = \frac{2Q}{C_1\omega_0(4Q^2 + 1)}$$

The gain K must be unity, but Q can be up to 10. Using these relations, compute the appropriate values of the resistors to achieve center frequency f_0 = 1 KHz, Gain K = 1 and Q = 10.

Choose the capacitors as $C_1 = C_2 = 0.1\ \mu F$ and $C_3 = 2C_1$. Plot |G(s)| versus frequency.

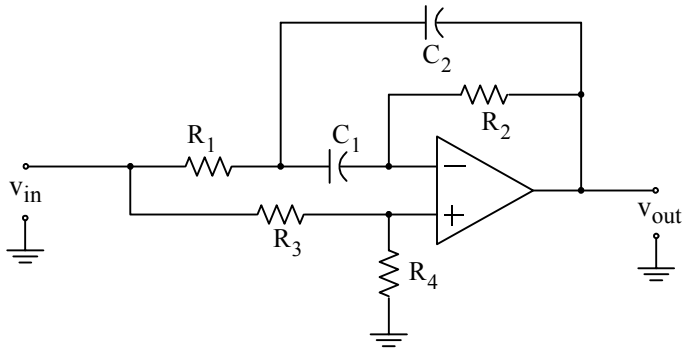
Solution using MATLAB is highly recommended.

4. The op amp circuit below is a MFB second–order *all–pass filter* or *phase shift filter* whose transfer function is

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{K(s^2 - a\omega_0 s + b\omega_0^2)}{s^2 + a\omega_0 s + b\omega_0^2}$$

where the gain $K = \text{constant}$, $(0 < K < 1)$, and the phase is given by

$$\phi(\omega) = -2\tan^{-1}\left(\frac{a\omega_0\omega}{b\omega_0^2 - \omega^2}\right)$$



The coefficients a and b can be found from

$$\phi_0 = \phi(\omega_0) = -2\tan^{-1}\left(\frac{a}{b-1}\right)$$

For arbitrary values of $C_1 = C_2$, we can compute the resistances from

$$R_2 = \frac{2}{a\omega_0 C_1}$$

$$R_1 = \frac{(1-K)R_2}{4K}$$

$$R_3 = \frac{R_2}{K}$$

$$R_4 = \frac{R_2}{1-K}$$

For $0 < \phi_0 < 180°$, we compute the coefficient a from

$$a = \frac{1-K}{2K\tan(\phi_0/2)}\left[-1 + \sqrt{1 + \frac{4K}{1-K} \cdot tan^2(\phi_0/2)}\right]$$

and for $-180° < \phi_0 < 0°$, from

$$a = \frac{1-K}{2K\tan(\phi_0/2)}\left[-1-\sqrt{1+\frac{4K}{1-K}\cdot\tan^2(\phi_0/2)}\right]$$

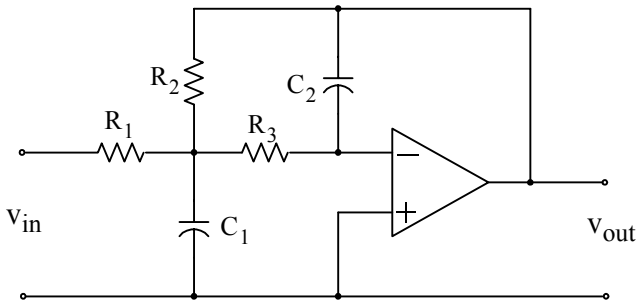
Using these relations, compute the appropriate values of the resistors to achieve a phase shift $\phi_0 = -90°$ at $f_0 = 1$ KHz with $K = 0.75$.

Choose the capacitors as $C_1 = C_2 = 0.01\ \mu F$ and plot phase versus frequency.

Solution using MATLAB is highly recommended.

5. The op amp circuit below is also known as *Bessel filter* and has the same configuration as the low–pass filter presented in Figure 4.20, Chapter 4, Page 4–15. This circuit achieves a relatively constant time delay over a range $0 < \omega < \omega_0$. The second-order transfer function of this filter is

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{3K\omega_0^2}{s^2+3\omega_0 s+3\omega_0^2}$$



where K is the gain and the time delay T_0 at $\omega_0 = 2\pi f_0$ is given as

$$T_0 = T(\omega_0) = \frac{12}{13\omega_0}\ \text{seconds}$$

We recognize the transfer function $|G(s)|$ above as that of a low–pass filter where $a = b = 3$ and the substitution of $\omega_0 = \omega_C$. Therefore, we can use a low–pass filter circuit such as that above to achieve a constant delay T_0 by specifying the resistor and capacitor values of the circuit.

The resistor values are computed from

$$R_2 = \frac{2(K+1)}{\left(aC_1+\sqrt{a^2C_1^2-4bC_1C_2(K-1)}\right)\omega_0} \qquad R_1 = \frac{R_2}{K} \qquad R_3 = \frac{1}{bC_1C_2R_2\omega_0^2}$$

Using these relations, compute the appropriate values of the resistors to achieve a time delay

$T_0 = 100\ \mu s$ with $K = 2$. Use capacitors $C_1 = 0.01\ \mu F$ and $C_2 = 0.002\ \mu F$. Plot $|G(s)|$ versus frequency.

Solution using MATLAB is highly recommended.

6. Derive the transfer function of a fourth–order Butterworth filter with $\omega_C = 1\ rad/s$.

7. Derive the magnitude–squared function for a third–order Chebyshev Type I low–pass filter with 1.5 dB pass band ripple and cutoff frequency $\omega_C = 1\ rad/s$.

8. In Chapter 4, Exercise 5, Page 4–30, for the RC low–pass filter we derived the transfer function

$$G(s) = \frac{1/RC}{s + 1/RC}$$

Derive the equivalent digital filter transfer function by application of the bilinear transformation. Assume that $RC = 1$

9. Use MATLAB to derive the transfer function $G(z)$ and plot $|G(z)|$ versus ω for a two–pole, Chebyshev Type I high–pass digital filter with sampling period $T_S = 0.25\ s$. The equivalent analog filter cutoff frequency is $\omega_C = 4\ rad/s$ and has 3 dB pass band ripple. Compute the coefficients of the numerator and denominator and plot $|G(z)|$ with and without pre–warping.

11.9 Solutions to End–of–Chapter Exercises

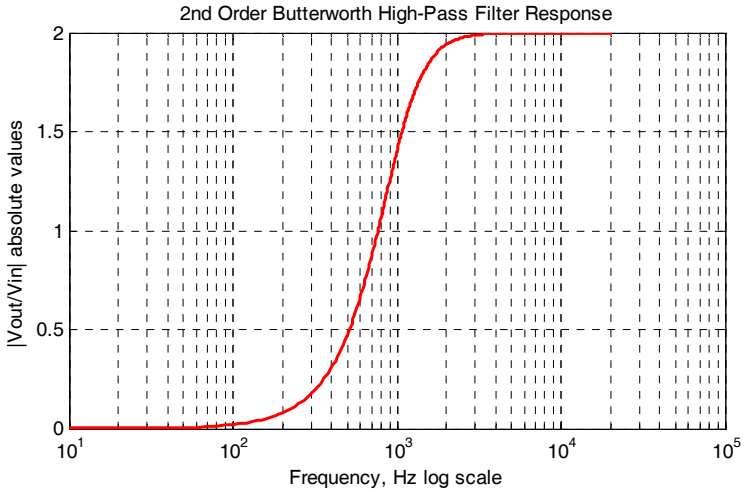
1. We will use MATLAB for all computations.

```
% PART I – Find Resistor values for second order Butterworth filter, a=sqrt(2), b = 1
%
a=sqrt(2); b =1; C1=10^(–8); C2=C1; fc=1000; wc=2*pi*fc; K=2;
R2=(4*b)/(C1*sqrt(a^2+8*b*(K–1))*wc);
R1=b/(C1^2*R2*wc^2); R3=(K*R2)/(K–1); R4=K*R2; fprintf(' \n');...
fprintf('R1 = %5.0f Ohms \t',R1); fprintf('R2 = %5.0f Ohms \t',R2);...
fprintf('R3 = %5.0f Ohms \t',R3); fprintf('R4 = %5.0f Ohms \t',R4)
```

```
R1=12582 Ohms   R2=20132 Ohms   R3=40263 Ohms   R4=40263 Ohms
```

We choose standard resistors as close as possible to those found above. These are shown in the MATLAB script below.

```
% PART II – Plot with standard resistors R1=12.7 K, R2=20.0 K, R3=40.2 K, R4= R3
%
f=10:10:20000; w=2*pi*f; R1=12700; R2=20000; R3=40200; R4=R3; K=1+R4/R3;...
wc=(4*b)/(C1*sqrt(a^2+8*b*(K–1))*R2); s=w*j; Gw=(K.*s.^2)./(s.^2+a.*wc.*s./b+wc.^2./b);...
semilogx(f,abs(Gw)); xlabel('Frequency, Hz log scale'), ylabel('|Vout/Vin| absolute values');...
title('2nd Order Butterworth High–Pass Filter Response'); grid
```



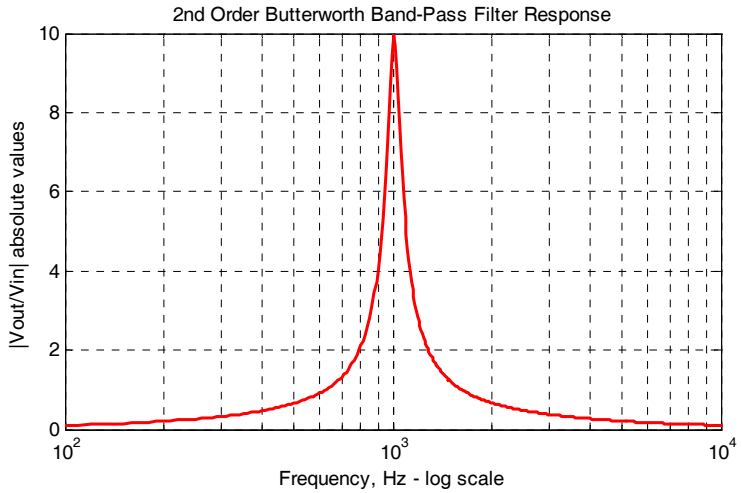
2. We will use MATLAB for all computations.

```
% PART I – Find Resistor values for second order band–pass filter f0 = 1 KHz
%
Q=10; K=10; C1=10^(–7); C2=C1; f0=1000; w0=2*pi*f0; R1=(2*Q)/(C1*w0*K);...
R2=(2*Q)/(C1*w0*(–1+sqrt((K–1)^2+8*Q^2))); R3=(1/(C1^2*w0^2))*(1/R1+1/R2); R4=2*R3;...
R5=R4; fprintf('  \n'); fprintf('R1 = %5.0f Ohms \t',R1); fprintf('R2 = %5.0f Ohms \t',R2);...
fprintf('R3 = %5.0f Ohms \t',R3); fprintf('R4 = %5.0f Ohms \t',R4);...
fprintf('R5 = %5.0f Ohms \t',R5)
```

```
R1=3183 Ohms R2=1110 Ohms R3=3078 Ohms R4=6156 Ohms R5=6156 Ohms
```

We choose standard resistors as close as possible to those found above. These are shown in the MATLAB script below.

```
%
% PART II – Plot with standard resistors R1=3.16 K, R2=1.1 K, R3=3.09 K, R4= 6.19 K,
% R5=R4
%
K=10; Q=10; f=10:10:10000; w=2*pi*f; R1=3160; R2=1100; R3=3090; R4=6190; R5=R4;...
w0=(2*Q)/(C1*R1*K); B=w0/Q; s=w*j; Gw=(K.*B.*s)./(s.^2+B.*s+w0.^2);...
semilogx(f,abs(Gw)); axis([100 10000 0 10]); xlabel('Frequency, Hz – log scale'),...
ylabel('|Vout/Vin| absolute values');...
title('2nd Order Butterworth Band–Pass Filter Response'); grid
```



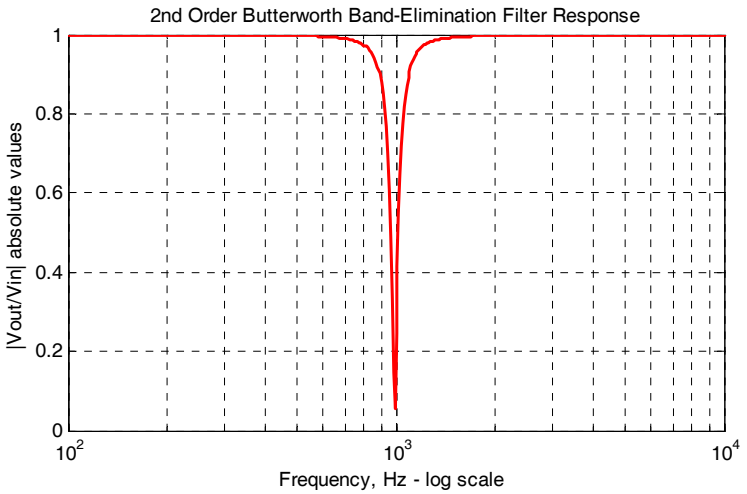
3. We will use MATLAB for all computations.

```
% PART I – Find Resistor values for second order Butterworth band–elimination filter
% with f0 = 1 KHz
%
Q=10; K=1; C1=10^(–7); C2=C1; C3=2*C1; f0=1000; w0=2*pi*f0;...
R1=1/(2*w0*Q*C1); R2=(2*Q)/(w0*C1); R3=(2*Q)/(C1*w0*(4*Q^2+1)); fprintf(' \n');...
fprintf('R1 = %5.0f Ohms \t',R1); fprintf('R2 = %5.0f Ohms \t',R2);...
fprintf('R3 = %5.0f Ohms \t',R3)
```

```
R1=80 Ohms   R2=31831 Ohms   R3=79 Ohms
```

We choose standard resistors as close as possible to those found above. These are shown in the MATLAB script below.

```
%
% PART II – Plot with standard resistors R1=80.6, R2=3.16 K, R3=78.7
%
K=1; Q=10; f=10:10:10000; w=2*pi*f; R1=80.6; R2=31600; R3=78.7;...
w0=1/(2*R1*Q*C1); B=w0/Q; s=w*j; Gw=(K.*(s.^2+w0.^2))./(s.^2+B.*s+w0.^2);...
semilogx(f,abs(Gw)); axis([100 10000 0 1]); xlabel('Frequency, Hz – log scale ');...
ylabel('|Vout/Vin| absolute values');...
title('2nd Order Butterworth Band–Elimination Filter Response'); grid
```



4. Let us first solve the relation

$$\phi_0 = \phi(\omega_0) = -2\tan^{-1}\left(\frac{a}{b-1}\right)$$

for b in terms of ϕ_0 and a so that we can derive its value from the MATLAB script below. We rewrite the above relation as

$$\tan^{-1}\left(\frac{a}{b-1}\right) = -\frac{\phi_0}{2}$$

then,

$$\tan\left(-\frac{\phi_0}{2}\right) = \frac{a}{b-1}$$

$$b\tan\left(-\frac{\phi_0}{2}\right) = a + \tan\left(-\frac{\phi_0}{2}\right)$$

$$b = \frac{a + \tan(-\phi_0/2)}{\tan(-\phi_0/2)}$$

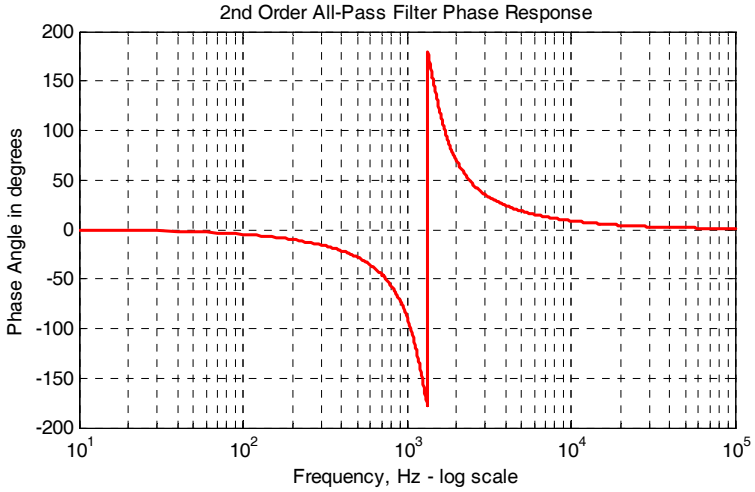
```
% MFB 2nd order all–pass filter, f0=1 KHz
% phase shift phi=–pi/2 and gain K=3/4. Gain must be 0<K<1
%
% PART I – Find Resistor, a and b values
%
phi0=–pi/2; K=0.75; C1=10^(–8); C2=C1; f0=1000; w0=2*pi*f0;...
a=((1–K)/(2*K*tan(phi0/2)))*(–1–sqrt((1+4*K/(1–K))*(tan(phi0/2))^2));...
b=(a+tan(–phi0/2))/tan(–phi0/2); R2=2/(a*w0*C1); R1=(1–K)*R2/(4*K);...
R3=R2/K; R4=R2/(1–K); fprintf('  \n'); fprintf('R1 = %6.0f Ohms \t',R1);...
fprintf('R2 = %6.0f Ohms \t',R2); fprintf('R3 = %6.0f Ohms \t',R3);...
fprintf('R4 = %6.0f Ohms \t',R4); fprintf('  \n');...
fprintf('a = %5.3f  \t', a); fprintf('b = %5.3f  \t', b)
```

```
R1=3456 Ohms R2=41469 Ohms  R3=55292 Ohms  R4=165875 Ohms
a = 0.768  b = 1.768
```

We choose standard resistors as close as possible to those found above. These are shown in the MATLAB script below.

```
%
% PART II – Plot with standard resistors R1=3.48 K, R2=41.2 K, R3=54.9 K, R4=165 K
%
K=3/4; a=0.768; b=1.768; C1=10^(–8); C2=C1; f=10:10:100000; w=2*pi*f;...
R1=3480; R2=41200; R3=54900; R4=165000; w0=2/(a*R2*C1); s=w*j;
Gw=(K.*(s.^2–a.*w0.*s+b.*w0.^2))./(s.^2+a.*w0.*s+b.*w0.^2);...
semilogx(f,angle(Gw).*180./pi); xlabel('Frequency, Hz – log scale');...
ylabel('Phase Angle in degrees'); title('2nd Order All–Pass Filter Phase Response'); grid
```

The plot shown below is the phase (not magnitude) response.



5. We will use MATLAB for all computations.

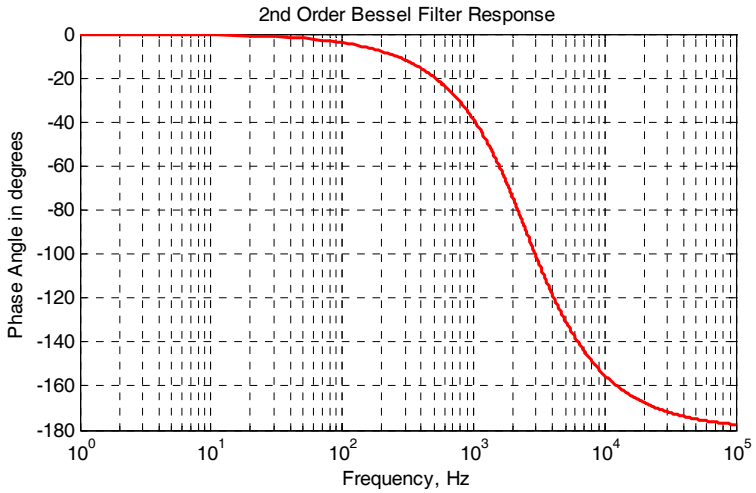
```
% MFB 2nd order Bessel filter, T0=100 microseconds, K=2
% PART I – Find resistor values
T0=100*10^(–6); K=2; C1=10^(–8); C2=2*10^(–9); a=3; b=3; w0=12/(13*T0);...
R2=(2*(K+1))/((a*C1+sqrt(a^2*C1^2–4*b*C1*C2*(K+1)))*w0); R1=R2/K;...
R3=1/(b*C1*C2*R2*w0^2); fprintf(' \n'); fprintf('R1 = %5.0f Ohms \t',R1);...
fprintf('R2 = %5.0f Ohms \t',R2); fprintf('R3 = %5.0f Ohms \t',R3)
```

```
R1 = 7486 Ohms    R2 = 14971 Ohms    R3 = 13065 Ohms
```

We choose standard resistors as close as possible to those found above. These are shown in the MATLAB script below. Part II of the script is as follows:

```
%
% PART II – Plot with standard resistors R1=7.5 K, R2=15.0 K, R3=13.0 K
%
K=2; a=3; b=3; C1=10^(–8); C2=2*10^(–9); f=1:10:100000; w=2*pi*f; R1=7500;...
R2=15000; R3=13000; w0=(2*(K+1))/((a*C1+sqrt(a^2*C1^2–4*b*C1*C2*(K+1)))*R2);...
s=w*j; Gw=(3.*K.*w0.^2)./(s.^2+a.*w0.*s+b.*w0.^2);...
semilogx(f,angle(Gw).*180./pi); xlabel('Frequency, Hz');...
ylabel('Phase Angle in degrees'); title('2nd Order Bessel Filter Response'); grid
```

The plot shown below is the phase (not magnitude) response. This filter has very good phase response but poor magnitude response. The group delay (the slope at a particular frequency) is practically flat at frequencies near DC.



6. From (11.24), Page 11–14,

$$A^2(\omega) = \frac{1}{(\omega/\omega_C)^{2k} + 1}$$

and with $\omega_C = 1$ rad/s and $k = 4$, we obtain

$$A^2(\omega) = \frac{1}{\omega^8 + 1}$$

Then,

$$G(s) \cdot G(-s) = \frac{1}{s^8 + 1}$$

We can use DeMoivre's theorem to find the roots of $s^8 + 1$ but we will use MATLAB instead.

```
syms s; y=solve('s^8+1=0'); fprintf(' \n'); disp('s1 = '); disp(simple(y(1)));...
disp('s2 = '); disp(simple(y(2))); disp('s3 = '); disp(simple(y(3)));...
disp('s4 = '); disp(simple(y(4))); disp('s5 = '); disp(simple(y(5)));...
disp('s6 = '); disp(simple(y(6))); disp('s7 = '); disp(simple(y(7)));...
disp('s8 = '); disp(simple(y(8)))
```

```
s1 = 1/2*2^(3/4)*(1+i)^(1/2)
```

$$s_1 = (1/2) \cdot \sqrt[4]{2^3} \cdot \sqrt{1+j}$$

```
s2 = -1/2*2^(3/4)*(1+i)^(1/2)
```

$$s_2 = -(1/2) \cdot \sqrt[4]{2^3} \cdot \sqrt{1+j}$$

```
s3 = 1/2*i*2^(3/4)*(1+i)^(1/2)
```

$$s_3 = (1/2)j \cdot \sqrt[4]{2^3} \cdot \sqrt{1+j}$$

```
s4 = -1/2*i*2^(3/4)*(1+i)^(1/2)
```

$$s_4 = -(1/2)j \cdot \sqrt[4]{2^3} \cdot \sqrt{1+j}$$

```
s5 = 1/2*i*2^(3/4)*(-1+i)^(1/2)
```

$$s_5 = (1/2)j \cdot \sqrt[4]{2^3} \cdot \sqrt{-1+j}$$

```
s6 = -1/2*i*2^(3/4)*(-1+i)^(1/2)
```
$$s_6 = -(1/2)j \cdot \sqrt[4]{2^3} \cdot \sqrt{-1+j}$$

```
s7 = 1/2*2^(3/4)*(-1+i)^(1/2)
```
$$s_7 = (1/2) \cdot \sqrt[4]{2^3} \cdot \sqrt{-1+j}$$

```
s8 = -1/2*2^(3/4)*(-1+i)^(1/2)
```
$$s_8 = -(1/2) \cdot \sqrt[4]{2^3} \cdot \sqrt{-1+j}$$

Since we are only interested in the poles of the left half of the s–plane, we choose the roots s_2, s_4, s_6, and s_8. To express the denominator in polynomial form we use the following MATLAB script:

```
denGs=(s-s2)*(s-s4)*(s-s6)*(s-s8); r=vpa(denGs,4)
r = (s+.9240+.3827*i)*(s+.3827-.9240*i)*(s+.9240-.3827*i)*(s+.3827+.9240*i)
expand(r)

ans =
s^4+2.6134*s^3+3.41492978*s^2+2.614014906886*s
  +1.0004706353613841
```

and thus

$$G(s) = \frac{1}{s^4 + 2.61s^3 + 3.41s^2 + 2.61s + 1}$$

7. From (11.49), Page 11–26,

$$A^2(\omega) = \frac{\alpha}{1 + \varepsilon^2 C_k^2(\omega/\omega_C)} \quad (1)$$

and with $\omega_C = 1$ and $k = 3$, we find from (11.48), Page 11–26, that

$$C_k^2 = C_3^2 = (4\omega^3 - 3\omega)^2$$

Also, from (11.56), Page 11–29, with 1.5 dB ripple,

$$\varepsilon^2 = 10^{(1.5/10)} - 1 = 0.4125$$

and with these values (1) is written as

$$A^2(\omega) = \frac{\alpha}{1 + 0.4125 \cdot (4\omega^3 - 3\omega)^2}$$

To express the denominator in polynomial form, we use the following MATLAB script.

```
syms w; denA=1+0.4125*expand((4*w^3-3*w)^2);...
denA = 1+33/5*w^6-99/10*w^4+297/80*w^2
```

and thus

$$A^2(\omega) = \frac{\alpha}{6.6\omega^6 - 9.9\omega^4 + 3.7125\omega^2 + 1}$$

8.

$$G(s) = \frac{1/RC}{s + 1/RC}$$

and with $RC = 1$,

$$G(s) = \frac{1}{s+1}$$

$$G(z) = G(s)\Big|_{s = \frac{2}{T_S}\cdot\frac{z-1}{z+1}}$$

$$G(z) = \frac{1}{\frac{2}{T_S}\cdot\frac{z-1}{z+1} + 1} = \frac{T_S(z+1)}{2(z-1) + T_S(z+1)} = \frac{T_S(z+1)}{(T_S+2)z + (T_S-2)} = \frac{T_S(z+1)/(T_S+2)}{z + (T_S-2)/(T_S+2)}$$

9. The approximation of (11.86), Page 11–54, with $\omega_c = \omega_a$ yields $\omega_c = \omega_a \cdot T_S = 4 \times 0.25 = 1$. However, using the exact relation of (11.85), Page 11–54, and solving for ω_d we find that

$$\omega_d = 2\tan^{-1}(\omega_a T_S)/2 = 0.9273$$

and this value is not very close to unity. Therefore we will compute $G_1(z)$ with pre–warping using the following MATLAB script.

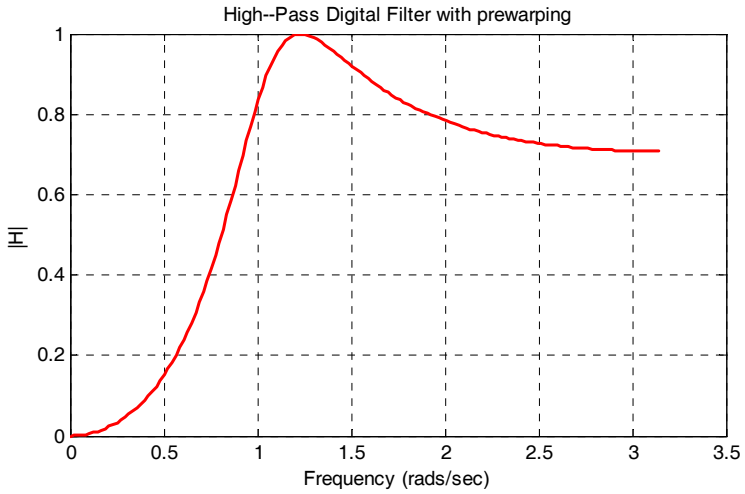
```
% This script designs a 2–pole Chebyshev Type 1 high–pass digital filter with
% analog cutoff frequency wc=4 rads/sec, sampling period Ts=0.25 sec., with
% pass band % ripple of 3 dB.
%
N=2;                    % # of poles
Rp=3;                   % Pass–band ripple in dB
Ts=0.25;                % Sampling period
wc=4;                   % Analog cutoff frequency
% Let wd be the discrete time radian frequency. This frequency is related to
% the continuous time radian frequency wc by wd=Ts*wc with no pre–warping.
% With prewarping it is related to wc by wdp=2*arctan(wc*Ts/2).
% We divide by pi to normalize the digital cutoff frequency.
wdp=2*atan(wc*Ts/2)/pi;
% To obtain the digital cutoff frequency without prewarping we use the relation
% wd=(wc*Ts)/pi;
[Nz,Dz]=cheby1(N,Rp,wdp,'high');
%
fprintf('The numerator N(z) coefficients in descending powers of z are: \n\n');
fprintf('%8.4f \t',[Nz]); fprintf(' \n');
```

```
fprintf('The denominator D(z) coefficients in descending powers of z are: \n\n');
fprintf('%8.4f \t',[Dz]); fprintf(' \n');
%
fprintf('Press any key to see the plot \n');
pause;
%
w=0:2*pi/300:pi; Gz=freqz(Nz,Dz,w); plot(w,abs(Gz)); grid; xlabel('Frequency (rads/sec)');
ylabel('|H|'); title('High–Pass Digital Filter with pre–warping')
```

```
The numerator N(z) coefficients in descending powers of z are:
 0.3914   -0.7829    0.3914

The denominator D(z) coefficients in descending powers of z
are:

  1.0000  -0.7153   0.4963
```

and thus the transfer function and the plot with pre–warping are as shown below.

$$G_1(z) = \frac{0.3914z^2 - 0.7829z + 0.3914}{z^2 - 0.7153z + 0.4963}$$



Next, we will compute $G_2(z)$ without pre–warping using the following MATLAB script:

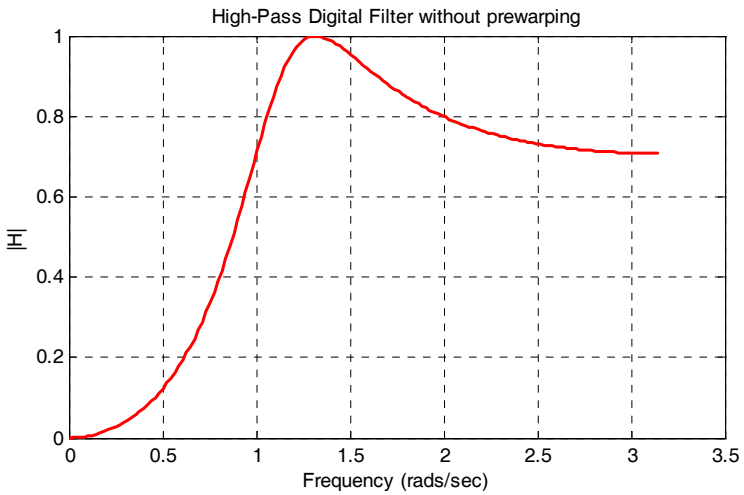
```
N=2;                        % # of poles
Rp=3;                       % Pass band ripple in dB
Ts=0.25;                    % Sampling period
wc=4;                       % Analog cutoff frequency
wd=(wc*Ts)/pi;
[Nz,Dz]=cheby1(N,Rp,wd,'high');
%
```

```
fprintf('The numerator N(z) coefficients in descending powers of z are: \n\n');
fprintf('%8.4f \t',[Nz]); fprintf(' \n');
fprintf('The denominator D(z) coefficients in descending powers of z are: \n\n');
fprintf('%8.4f \t',[Dz]); fprintf(' \n');
%
fprintf('Press any key to see the plot \n');
pause;
%
w=0:2*pi/300:pi; Gz=freqz(Nz,Dz,w); plot(w,abs(Gz)); grid; xlabel('Frequency (rads/sec)');
ylabel('|H|'); title('High-Pass Digital Filter without pre-warping')
```

```
The numerator N(z) coefficients in descending powers of z are:

   0.3689   -0.7377    0.3689

The denominator D(z) coefficients in descending powers of z
are:

   1.0000   -0.6028    0.4814
```

and thus the transfer function and the plot without pre-warping are as shown below.

$$G_2(z) = \frac{0.3689z^2 - 0.7377z + 0.3689}{z^2 - 0.6028z + 0.4814}$$



Appendix A

Introduction to MATLAB®

This appendix serves as an introduction to the basic MATLAB commands and functions, procedures for naming and saving the user generated files, comment lines, access to MATLAB's Editor / Debugger, finding the roots of a polynomial, and making plots. Several examples are provided with detailed explanations.

A.1 MATLAB® and Simulink®

MATLAB and Simulink are products of The MathWorks,™ Inc. These are two outstanding software packages for scientific and engineering computations and are used in educational institutions and in industries including automotive, aerospace, electronics, telecommunications, and environmental applications. MATLAB enables us to solve many advanced numerical problems rapidly and efficiently.

A.2 Command Window

To distinguish the screen displays from the user commands, important terms, and MATLAB functions, we will use the following conventions:

Click: Click the left button of the mouse
`Courier Font:` Screen displays
Helvetica Font: User inputs at MATLAB's command window prompt >> or `EDU>>`*
Helvetica Bold: MATLAB functions

Times Bold Italic: Important terms and facts, notes and file names
When we first start MATLAB, we see various help topics and other information. Initially, we are interested in the *command screen* which can be selected from the Window drop menu. When the command screen, we see the prompt >> or `EDU>>`. This prompt is displayed also after execution of a command; MATLAB now waits for a new command from the user. It is highly recommended that we use the *Editor/Debugger* to write our program, save it, and return to the command screen to execute the program as explained below.

To use the Editor/Debugger:

1. From the *File* menu on the toolbar, we choose *New* and click on M–*File*. This takes us to the *Editor Window* where we can type our *script* (list of statements) for a new file, or open a previously saved file. We must save our program with a file name which starts with a letter. ***Impor-***

* EDU>> *is the MATLAB prompt in the Student Version*

tant! MATLAB is *case sensitive*, that is, it distinguishes between upper– and lower–case letters. Thus, *t* and *T* are two different letters in MATLAB language. The files that we create are saved with the file name we use and the extension ***.m***; for example, ***myfile01.m***. It is a good practice to save the script in a file name that is descriptive of our script content. For instance, if the script performs some matrix operations, we ought to name and save that file as ***matrices01.m*** or any other similar name. We should also use a floppy disk or an external drive to backup our files.

2. Once the script is written and saved as an *m–file*, we may exit the *Editor/Debugger* window by clicking on *Exit Editor/Debugger* of the *File* menu. MATLAB then returns to the command window.

3. To execute a program, we type the file name ***without*** the ***.m*** extension at the >> prompt; then, we press <enter> and observe the execution and the values obtained from it. If we have saved our file in drive ***a*** or any other drive, we must make sure that it is added it to the desired directory in MATLAB's search path. The MATLAB User's Guide provides more information on this topic.

Henceforth, it will be understood that each input command is typed after the >> prompt and followed by the <enter> key.

The command **help matlab\iofun** will display input/output information. To get help with other MATLAB topics, we can type help followed by any topic from the displayed menu. For example, to get information on graphics, we type help matlab\graphics. The MATLAB User's Guide contains numerous help topics.

To appreciate MATLAB's capabilities, we type demo and we see the `MATLAB Demos` menu. We can do this periodically to become familiar with them. Whenever we want to return to the command window, we click on the `Close` button.

When we are done and want to leave MATLAB, we type **quit** or **exit**. But if we want to clear all previous values, variables, and equations without exiting, we should use the command **clear**. This command erases everything; it is like exiting MATLAB and starting it again. The command **clc** clears the screen but MATLAB still remembers all values, variables and equations that we have already used. In other words, if we want to clear all previously entered commands, leaving only the >> prompt on the upper left of the screen, we use the **clc** command.

All text after the **%** (percent) symbol is interpreted as a *comment line* by MATLAB, and thus it is ignored during the execution of a program. A comment can be typed on the same line as the function or command or as a separate line. For instance,

```
conv(p,q)    % performs multiplication of polynomials p and q

% The next statement performs partial fraction expansion of p(x) / q(x)
```

are both correct.

One of the most powerful features of MATLAB is the ability to do computations involving *complex numbers*. We can use either `i`, or `j` to denote the imaginary part of a complex number, such as `3-4i` or `3-4j`. For example, the statement

```
z=3–4j
```

displays

```
z = 3.0000–4.0000i
```

In the above example, a multiplication (*) sign between *4* and `j` was not necessary because the complex number consists of numerical constants. However, if the imaginary part is a function, or variable such as `cos(x)`, we must use the multiplication sign, that is, we must type **cos(x)*j** or **j*cos(x)** for the imaginary part of the complex number.

A.3 Roots of Polynomials

In MATLAB, a polynomial is expressed as a *row vector* of the form $[a_n \;\; a_{n-1} \;\; \ldots \;\; a_2 \;\; a_1 \;\; a_0]$. These are the coefficients of the polynomial in descending order. ***We must include terms whose coefficients are zero.***

We find the roots of any polynomial with the **roots(p)** function; **p** is a row vector containing the polynomial coefficients in descending order.

Example A.1

Find the roots of the polynomial

$$p_1(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$$

Solution:

The roots are found with the following two statements where we have denoted the polynomial as **p1**, and the roots as **roots_ p1**.

```
p1=[1 –10 35 –50 24]      % Specify and display the coefficients of p1(x)
p1 =
     1   -10    35   -50    24
roots_ p1=roots(p1)       % Find the roots of p1(x)
roots_p1 =
   4.0000
   3.0000
   2.0000
   1.0000
```

We observe that MATLAB displays the polynomial coefficients as a row vector, and the roots as a column vector.

Example A.2

Find the roots of the polynomial

$$p_2(x) = x^5 - 7x^4 + 16x^2 + 25x + 52$$

Solution:

There is no cube term; therefore, we must enter zero as its coefficient. The roots are found with the statements below, where we have defined the polynomial as **p2**, and the roots of this polynomial as **roots_ p2**. The result indicates that this polynomial has three real roots, and two complex roots. Of course, complex roots always occur in *complex conjugate** pairs.

```
p2=[1 –7  0  16  25  52]

p2 =
     1     –7      0     16     25     52
```

```
roots_ p2=roots(p2)

roots_p2 =
   6.5014
   2.7428
  –1.5711
  –0.3366 + 1.3202i
  –0.3366 – 1.3202i
```

A.4 Polynomial Construction from Known Roots

We can compute the coefficients of a polynomial, from a given set of roots, with the **poly(r)** function where **r** is a row vector containing the roots.

Example A.3

It is known that the roots of a polynomial are $1, 2, 3,$ and 4. Compute the coefficients of this polynomial.

* *By definition, the conjugate of a complex number* $A = a + jb$ *is* $A^* = a - jb$

Solution:

We first define a row vector, say r3 , with the given roots as elements of this vector; then, we find the coefficients with the **poly(r)** function as shown below.

```
r3=[1  2  3  4]          %  Specify the roots of the polynomial

r3 =
     1     2     3     4

poly_r3=poly(r3)         %  Find the polynomial coefficients

poly_r3 =
     1   -10    35   -50    24
```

We observe that these are the coefficients of the polynomial $p_1(x)$ of Example A.1.

Example A.4

It is known that the roots of a polynomial are $-1,\ -2,\ -3,\ 4+j5,$ and $4-j5$. Find the coefficients of this polynomial.

Solution:

We form a row vector, say r4 , with the given roots, and we find the polynomial coefficients with the **poly(r)** function as shown below.

```
r4=[ –1   –2   –3   4+5j   4–5j ]

r4 =
  Columns 1 through 4
  -1.0000   -2.0000   -3.0000   -4.0000+ 5.0000i
  Column 5
  -4.0000- 5.0000i

poly_r4=poly(r4)

poly_r4 =
     1    14   100   340   499   246
```

Therefore, the polynomial is

$$p_4(x) = x^5 + 14x^4 + 100x^3 + 340x^2 + 499x + 246$$

A.5 Evaluation of a Polynomial at Specified Values

The **polyval(p,x)** function evaluates a polynomial p(x) at some specified value of the independent variable x.

Example A.5

Evaluate the polynomial

$$p_5(x) = x^6 - 3x^5 + 5x^3 - 4x^2 + 3x + 2 \tag{A.1}$$

at $x = -3$.

Solution:

```
p5=[1 –3  0  5 –4  3  2];  % These are the coefficients of the given polynomial
                           % The semicolon (;) after the right bracket suppresses the
                           %  display of the row vector that contains the coefficients of p5.
%
val_minus3=polyval(p5, –3)  % Evaluate p5 at x=–3; no semicolon is used here
                            % because we want the answer to be displayed

val_minus3 =
        1280
```

Other MATLAB functions used with polynomials are the following:

conv(a,b) – multiplies two polynomials **a** and **b**

[q,r]=deconv(c,d) –divides polynomial **c** by polynomial **d** and displays the quotient **q** and remainder **r**.

polyder(p) – produces the coefficients of the derivative of a polynomial **p**.

Example A.6

Let

$$p_1 = x^5 - 3x^4 + 5x^2 + 7x + 9$$

and

$$p_2 = 2x^6 - 8x^4 + 4x^2 + 10x + 12$$

Compute the product $p_1 \cdot p_2$ using the **conv(a,b)** function.

Solution:

```
p1=[1 –3  0 5 7 9];          % The coefficients of p1
p2=[2  0 –8 0 4 10 12];      % The coefficients of p2
p1p2=conv(p1,p2)             % Multiply p1 by p2 to compute coefficients of the product p1p2
```

```
p1p2 =
  2   -6   -8   34   18   -24   -74   -88   78   166   174   108
```

Therefore,

$$p_1 \cdot p_2 = 2x^{11} - 6x^{10} - 8x^9 + 34x^8 + 18x^7 - 24x^6$$
$$-74x^5 - 88x^4 + 78x^3 + 166x^2 + 174x + 108$$

Example A.7

Let

$$p_3 = x^7 - 3x^5 + 5x^3 + 7x + 9$$

and

$$p_4 = 2x^6 - 8x^5 + 4x^2 + 10x + 12$$

Compute the quotient p_3/p_4 using the **[q,r]=deconv(c,d)** function.

Solution:

```
% It is permissible to write two or more statements in one line separated by semicolons
p3=[1  0 –3  0  5  7  9]; p4=[2 –8  0  0  4 10 12]; [q,r]=deconv(p3,p4)
```

```
q =
    0.5000
r =
     0     4    -3     0     3     2     3
```

Therefore,

$$q = 0.5 \qquad r = 4x^5 - 3x^4 + 3x^2 + 2x + 3$$

Example A.8

Let

$$p_5 = 2x^6 - 8x^4 + 4x^2 + 10x + 12$$

Compute the derivative $\frac{d}{dx}p_5$ using the **polyder(p)** function.

Solution:

```
p5=[2  0  –8  0  4  10  12];    % The coefficients of p5
der_p5=polyder(p5)              % Compute the coefficients of the derivative of p5
```

```
der_p5 =
    12     0   -32     0     8    10
```

Therefore,

$$\frac{d}{dx}p_5 = 12x^5 - 32x^3 + 4x^2 + 8x + 10$$

A.6 Rational Polynomials

Rational Polynomials are those which can be expressed in ratio form, that is, as

$$R(x) = \frac{Num(x)}{Den(x)} = \frac{b_n x^n + b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \ldots + b_1 x + b_0}{a_m x^m + a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_1 x + a_0} \tag{A.2}$$

where some of the terms in the numerator and/or denominator may be zero. We can find the roots of the numerator and denominator with the **roots(p)** function as before.

As noted in the comment line of Example A.7, we can write MATLAB statements in one line, if we separate them by commas or semicolons. ***Commas will display the results whereas semicolons will suppress the display.***

Example A.9

Let

$$R(x) = \frac{p_{num}}{p_{den}} = \frac{x^5 - 3x^4 + 5x^2 + 7x + 9}{x^6 - 4x^4 + 2x^2 + 5x + 6}$$

Express the numerator and denominator in factored form, using the **roots(p)** function.

Solution:

```
num=[1  –3  0  5  7  9]; den=[1  0  –4  0  2  5  6];    % Do not display num and den coefficients
roots_num=roots(num), roots_den=roots(den)              % Display num and den roots
```

```
roots_num =
   2.4186 + 1.0712i    2.4186 - 1.0712i   -1.1633
  -0.3370 + 0.9961i   -0.3370 - 0.9961i
```

```
roots_den =
   1.6760 + 0.4922i      1.6760 - 0.4922i   -1.9304
  -0.2108 + 0.9870i     -0.2108 - 0.9870i   -1.0000
```

As expected, the complex roots occur in complex conjugate pairs.

For the numerator, we have the factored form

$$p_{num} = (x-2.4186-j1.0712)(x-2.4186+j1.0712)(x+1.1633)$$
$$(x+0.3370-j0.9961)(x+0.3370+j0.9961)$$

and for the denominator, we have

$$p_{den} = (x-1.6760-j0.4922)(x-1.6760+j0.4922)(x+1.9304)$$
$$(x+0.2108-j0.9870)(x+0.2108+j0.9870)(x+1.0000)$$

We can also express the numerator and denominator of this rational function as a combination of *linear* and *quadratic* factors. We recall that, in a quadratic equation of the form $x^2+bx+c=0$ whose roots are x_1 and x_2, the negative sum of the roots is equal to the coefficient b of the x term, that is, $-(x_1+x_2)=b$, while the product of the roots is equal to the constant term c, that is, $x_1 \cdot x_2 = c$. Accordingly, we form the coefficient b by addition of the complex conjugate roots and this is done by inspection; then we multiply the complex conjugate roots to obtain the constant term c using MATLAB as follows:

```
(2.4186 + 1.0712i)*(2.4186 -1.0712i)
ans = 6.9971
(-0.3370+ 0.9961i)*(-0.3370-0.9961i)
ans = 1.1058
(1.6760+ 0.4922i)*(1.6760-0.4922i)
ans = 3.0512
(-0.2108+ 0.9870i)*(-0.2108-0.9870i)
ans = 1.0186
```

Thus,

$$R(x) = \frac{p_{num}}{p_{den}} = \frac{(x^2-4.8372x+6.9971)(x^2+0.6740x+1.1058)(x+1.1633)}{(x^2-3.3520x+3.0512)(x^2+0.4216x+1.0186)(x+1.0000)(x+1.9304)}$$

We can check this result of Example A.9 above with MATLAB's *Symbolic Math Toolbox* which is a collection of tools (functions) used in solving symbolic expressions. They are discussed in detail in MATLAB's Users Manual. For the present, our interest is in using the **collect(s)** function that is used to multiply two or more symbolic expressions to obtain the result in polynomial form. We must remember that the **conv(p,q)** function is used with numeric expressions only, that is, polynomial coefficients.

Before using a symbolic expression, we must create one or more symbolic variables such as x, y, t, and so on. For our example, we use the following script:

```
syms x  % Define a symbolic variable and use collect(s) to express numerator in polynomial form
collect((x^2–4.8372*x+6.9971)*(x^2+0.6740*x+1.1058)*(x+1.1633))
```

```
ans =
  x^5-29999/10000*x^4-1323/3125000*x^3+7813277909/
  1562500000*x^2+1750276323053/250000000000*x+4500454743147/
  500000000000
```

and if we simplify this, we find that is the same as the numerator of the given rational expression in polynomial form. We can use the same procedure to verify the denominator.

A.7 Using MATLAB to Make Plots

Quite often, we want to plot a set of ordered pairs. This is a very easy task with the MATLAB **plot(x,y)** command that plots *y* versus *x,* where *x* is the horizontal axis (abscissa) and *y* is the vertical axis (ordinate).

Example A.10

Consider the electric circuit of Figure A.1, where the radian frequency ω (radians/second) of the applied voltage was varied from 300 to 3000 in steps of 100 radians/second, while the amplitude was held constant.

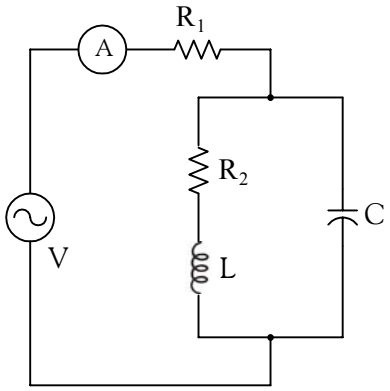


Figure A.1. Electric circuit for Example A.10

The ammeter readings were then recorded for each frequency. The magnitude of the impedance $|Z|$ was computed as $|Z| = |V/A|$ and the data were tabulated on Table A.1.

TABLE A.1 Table for Example A.10

ω (rads/s)	|Z| Ohms	ω (rads/s)	|Z| Ohms
300	39.339	1700	90.603
400	52.589	1800	81.088
500	71.184	1900	73.588
600	97.665	2000	67.513
700	140.437	2100	62.481
800	222.182	2200	58.240
900	436.056	2300	54.611
1000	1014.938	2400	51.428
1100	469.83	2500	48.717
1200	266.032	2600	46.286
1300	187.052	2700	44.122
1400	145.751	2800	42.182
1500	120.353	2900	40.432
1600	103.111	3000	38.845

Plot the magnitude of the impedance, that is, $|Z|$ versus radian frequency ω.

Solution:

We cannot type ω (omega) in the MATLAB Command prompt, so we will use the English letter w instead.

If a statement, or a row vector is too long to fit in one line, it can be continued to the next line by typing three or more periods, then pressing <*enter*> to start a new line, and continue to enter data. This is illustrated below for the data of w and z. Also, as mentioned before, we use the semicolon (;) to suppress the display of numbers that we do not care to see on the screen.

The data are entered as follows:

```
w=[300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900....
2000 2100 2200 2300 2400 2500 2600 2700 2800 2900 3000];
%
z=[39.339  52.789  71.104  97.665  140.437  222.182  436.056....
1014.938  469.830  266.032 187.052 145.751 120.353  103.111....
90.603  81.088  73.588  67.513  62.481  58.240  54.611  51.468....
48.717  46.286  44.122  42.182  40.432  38.845];
```

Of course, if we want to see the values of *w* or *z* or both, we simply type w or z, and we press

<*enter*>. To plot z (y–axis) versus w (x–axis), we use the **plot(x,y)** command. For this example, we use **plot(w,z)**. When this command is executed, MATLAB displays the plot on MATLAB's *graph screen* and MATLAB denotes this plot as Figure 1. This plot is shown in Figure A.2.

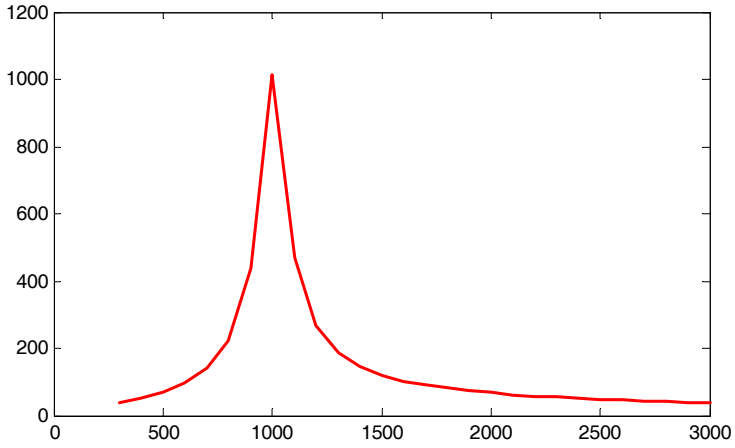


Figure A.2. Plot of impedance |z| *versus frequency* ω *for Example A.10*

This plot is referred to as the *magnitude frequency response* of the circuit.

To return to the command window, we press any key, or from the *Window* pull–down menu, we select *MATLAB Command Window*. To see the graph again, we click on the Window pull–down menu, and we choose *Figure 1*.

We can make the above, or any plot, more presentable with the following commands:

grid on: This command adds grid lines to the plot. The **grid off** command removes the grid. The command **grid** toggles them, that is, changes from off to on or vice versa. The default* is off.

box off: This command removes the box (the solid lines which enclose the plot), and **box on** restores the box. The command **box** toggles them. The default is on.

title('string'): This command adds a line of the text **string** (label) at the top of the plot.

xlabel('string') and **ylabel('string')** are used to label the x– and y–axis respectively.

The magnitude frequency response is usually represented with the x–axis in a logarithmic scale. We can use the **semilogx(x,y)** command which is similar to the **plot(x,y)** command, except that the x–axis is represented as a log scale, and the y–axis as a linear scale. Likewise, the **semilogy(x,y)** command is similar to the **plot(x,y)** command, except that the y–axis is represented as a

* *A default is a particular value for a variable that is assigned automatically by an operating system and remains in effect unless canceled or overridden by the operator.*

log scale, and the x–axis as a linear scale. The **loglog(x,y)** command uses logarithmic scales for both axes.

Throughout this text it will be understood that ***log*** is the common (base 10) logarithm, and ***ln*** is the natural (base e) logarithm. We must remember, however, the function **log(x)** in MATLAB is the natural logarithm, whereas the common logarithm is expressed as **log10(x)**, and the logarithm to the base 2 as **log2(x).**

Let us now redraw the plot with the above options by adding the following statements:

```
semilogx(w,z); grid;                % Replaces the plot(w,z) command
title('Magnitude of Impedance vs. Radian Frequency');
xlabel('w in rads/sec'); ylabel('|Z| in Ohms')
```

After execution of these commands, the plot is as shown in Figure A.3.

If the y–axis represents power, voltage or current, the x–axis of the frequency response is more often shown in a logarithmic scale, and the y–axis in dB (decibels).

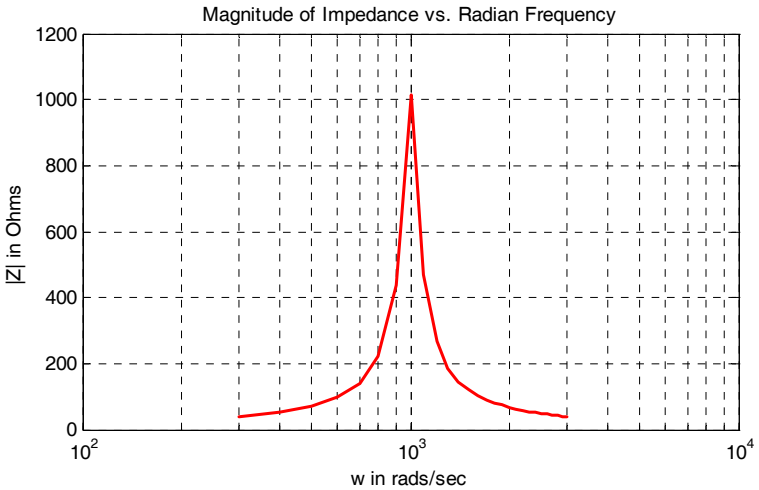


Figure A.3. Modified frequency response plot of Figure A.2.

To display the voltage v in a dB scale on the y–axis, we add the relation dB=20*log10(v), and we replace the **semilogx(w,z)** command with semilogx(w,dB).

The command **gtext('string')*** switches to the current *Figure Window*, and displays a cross–hair that can be moved around with the mouse. For instance, we can use the command gtext('Impedance |Z| versus Frequency'), and this will place a cross–hair in the *Figure* window. Then, using

* *With the latest MATLAB Versions 6 and 7 (Student Editions 13 and 14), we can add text, lines and arrows directly into the graph using the tools provided on the Figure Window. For advanced MATLAB graphics, please refer to The MathWorks* ***Using MATLAB Graphics*** *documentation.*

the mouse, we can move the cross–hair to the position where we want our label to begin, and we press <enter>.

The command **text(x,y,'string')** is similar to **gtext('string')**. It places a label on a plot in some specific location specified by **x** and **y**, and **string** is the label which we want to place at that location. We will illustrate its use with the following example which plots a *3–phase* sinusoidal waveform.

The first line of the script below has the form

linspace(first_value, last_value, number_of_values)

This function specifies ***the number of data points*** but not the increments between data points. An alternate function is

x=first: increment: last

and this specifies ***the increments between points*** but not the number of data points.

The script for the 3–phase plot is as follows:

```
x=linspace(0, 2*pi, 60);          %  pi is a built–in function in MATLAB;
%  we could have used x=0:0.02*pi:2*pi or x = (0: 0.02: 2)*pi instead;
y=sin(x); u=sin(x+2*pi/3); v=sin(x+4*pi/3);
plot(x,y,x,u,x,v);                %  The x–axis must be specified for each function
grid on, box on,                  %  turn grid and axes box on
text(0.75, 0.65, 'sin(x)');  text(2.85, 0.65, 'sin(x+2*pi/3)'); text(4.95, 0.65, 'sin(x+4*pi/3)')
```

These three waveforms are shown on the same plot of Figure A.4.

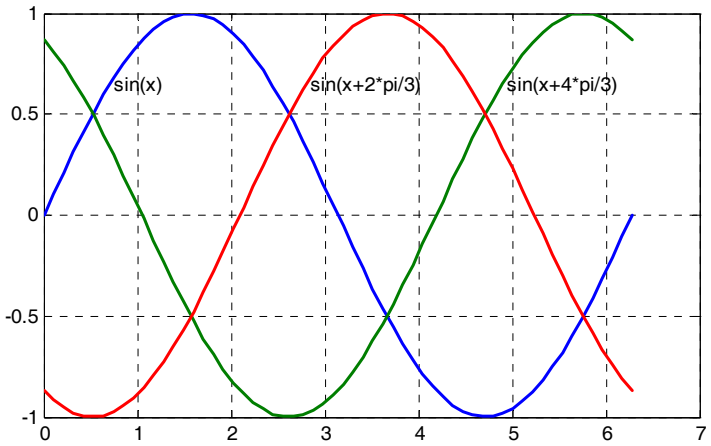


Figure A.4. Three–phase waveforms

In our previous examples, we did not specify line styles, markers, and colors for our plots. However, MATLAB allows us to specify various line types, plot symbols, and colors. These, or a combination of these, can be added with the **plot(x,y,s)** command, where **s** is a character string containing one or more characters shown on the three columns of Table A.2. MATLAB has no default color; it starts with blue and cycles through the first seven colors listed in Table A.2 for each additional line in the plot. Also, there is no default marker; no markers are drawn unless they are selected. The default line is the solid line. But with the latest MATLAB versions, we can select the line color, line width, and other options directly from the *Figure Window.*

TABLE A.2 Styles, colors, and markets used in MATLAB

Symbol	*Color*	*Symbol*	*Marker*	*Symbol*	*Line Style*
b	blue	.	point	–	solid line
g	green	o	circle	:	dotted line
r	red	x	x–mark	–.	dash–dot line
c	cyan	+	plus	––	dashed line
m	magenta	*	star		
y	yellow	s	square		
k	black	d	diamond		
w	white	∨	triangle down		
		∧	triangle up		
		<	triangle left		
		>	triangle right		
		p	pentagram		
		h	hexagram		

For example, **plot(x,y,'m*:')** plots a magenta dotted line with a star at each data point, and **plot(x,y,'rs')** plots a red square at each data point, but does not draw any line because no line was selected. If we want to connect the data points with a solid line, we must type **plot(x,y,'rs–')**. For additional information we can type **help plot** in MATLAB's command screen.

The plots we have discussed thus far are two–dimensional, that is, they are drawn on two axes. MATLAB has also a three–dimensional (three–axes) capability and this is discussed next.

The **plot3(x,y,z)** command plots a line in *3–space* through the points whose coordinates are the elements of *x, y* and *z*, where *x, y* and *z* are three vectors of the same length.

The general format is **plot3(x_1,y_1,z_1,s_1,x_2,y_2,z_2,s_2,x_3,y_3,z_3,s_3,...)** where $\mathbf{x_n}$, $\mathbf{y_n}$ and $\mathbf{z_n}$ are vectors or matrices, and $\mathbf{s_n}$ are strings specifying color, marker symbol, or line style. These strings are the same as those of the two–dimensional plots.

Example A.11

Plot the function

$$z = -2x^3 + x + 3y^2 - 1 \tag{A.3}$$

Solution:

We arbitrarily choose the interval (length) shown on the script below.

```
x= –10: 0.5: 10;          %  Length of vector x
y= x;                     % Length of vector y must be same as x
z= –2.*x.^3+x+3.*y.^2–1;  %  Vector z is function of both x and y*
plot3(x,y,z); grid
```

The three–dimensional plot is shown in Figure A.5.

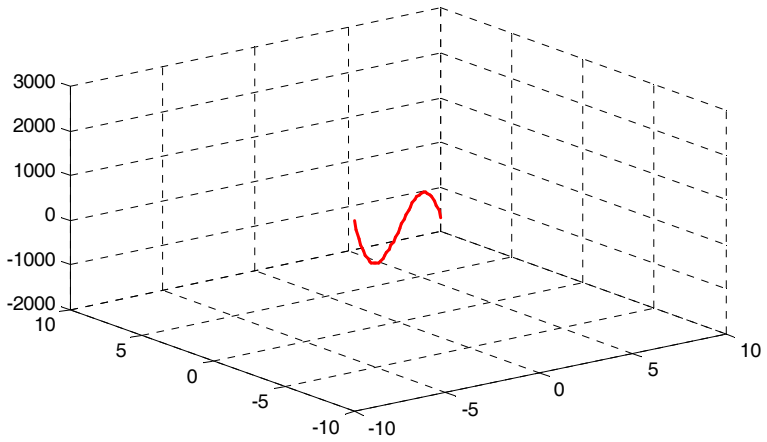


Figure A.5. Three dimensional plot for Example A.11

In a two–dimensional plot, we can set the limits of the *x*– and *y*–*axes* with the **axis([xmin xmax ymin ymax])** command. Likewise, in a three–dimensional plot we can set the limits of all three axes with the **axis([xmin xmax ymin ymax zmin zmax])** command. It must be placed after the **plot(x,y)** or **plot3(x,y,z)** commands, or on the same line without first executing the **plot** command. This must be done for each plot. The three–dimensional **text(x,y,z,'string')** command will place **string** beginning at the co–ordinate (*x*,*y*,*z*) on the plot.

For three–dimensional plots, **grid on** and **box off** are the default states.

* *This statement uses the so called dot multiplication, dot division, and dot exponentiation where the multiplication, division, and exponential operators are preceded by a dot. These important operations will be explained in Section A.9.*

We can also use the **mesh(x,y,z)** command with two vector arguments. These must be defined as length(x) = n and length(y) = m where [m, n] = size(Z). In this case, the vertices of the mesh lines are the triples {x(j), y(i), Z(i, j)}. We observe that **x** corresponds to the columns of Z, and **y** corresponds to the rows.

To produce a mesh plot of a function of two variables, say z = f(x, y), we must first generate the X and Y matrices that consist of repeated rows and columns over the range of the variables *x* and *y*. We can generate the matrices X and Y with the **[X,Y]=meshgrid(x,y)** function that creates the matrix X whose rows are copies of the vector **x**, and the matrix Y whose columns are copies of the vector **y**.

Example A.12

The volume V of a right circular cone of radius r and height h is given by

$$V = \frac{1}{3}\pi r^2 h \tag{A.4}$$

Plot the volume of the cone as r and h vary on the intervals $0 \le r \le 4$ and $0 \le h \le 6$ meters.

Solution:

The volume of the cone is a function of both the radius *r* and the height *h*, that is,

$$V = f(r, h)$$

The three–dimensional plot is created with the following MATLAB script where, as in the previous example, in the second line we have used the dot multiplication, dot division, and dot exponentiation. This will be explained in Section A.9.

```
[R,H]=meshgrid(0: 4, 0: 6);    % Creates R and H matrices from vectors r and h;...
V=(pi .* R .^ 2 .* H) ./ 3;  mesh(R, H, V);...
xlabel('x–axis, radius r (meters)'); ylabel('y–axis, altitude h (meters)');...
zlabel('z–axis, volume (cubic meters)'); title('Volume of Right Circular Cone'); box on
```

The three–dimensional plot of Figure A.6 shows how the volume of the cone increases as the radius and height are increased.

The plots of Figure A.5 and A.6 are rudimentary; MATLAB can generate very sophisticated three–dimensional plots. The MATLAB User's Manual and the Using MATLAB Graphics Manual contain numerous examples.

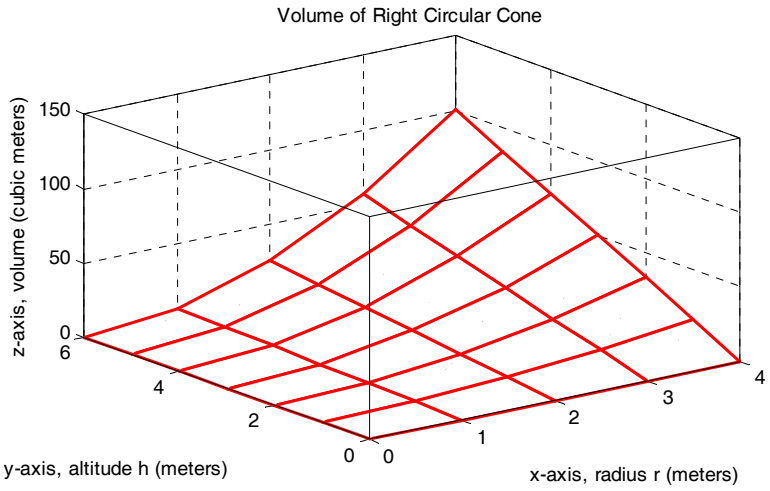


Figure A.6. Volume of a right circular cone.

A.8 Subplots

MATLAB can display up to four windows of different plots on the *Figure* window using the command **subplot(m,n,p)**. This command divides the window into an $m \times n$ matrix of plotting areas and chooses the *pth* area to be active. No spaces or commas are required between the three integers *m*, *n* and *p*. The possible combinations are shown in Figure A.7.

We will illustrate the use of the **subplot(m,n,p)** command following the discussion on multiplication, division and exponentiation that follows.

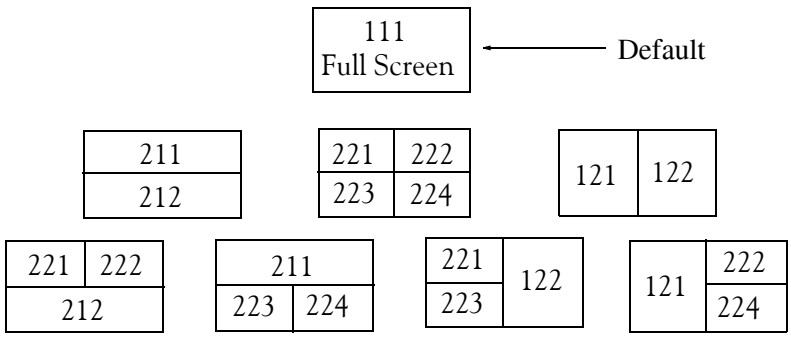


Figure A.7. Possible subplot arrangements in MATLAB

A.9 Multiplication, Division, and Exponentiation

MATLAB recognizes two types of multiplication, division, and exponentiation. These are the ***matrix*** multiplication, division, and exponentiation, and the ***element–by–element*** multiplication, division, and exponentiation. They are explained in the following paragraphs.

In Section A.2, the arrays [a b c …], such a those that contained the coefficients of polynomials, consisted of one row and multiple columns, and thus are called ***row vectors***. If an array has one column and multiple rows, it is called a ***column vector***. We recall that the elements of a row vector are separated by spaces. To distinguish between row and column vectors, the elements of a column vector must be separated by semicolons. An easier way to construct a column vector, is to write it first as a row vector, and then transpose it into a column vector. MATLAB uses the single quotation character (′) to transpose a vector. Thus, a column vector can be written either as

```
b=[–1; 3; 6; 11]
```

or as

```
b=[–1  3  6  11]'
```

As shown below, MATLAB produces the same display with either format.

```
b=[–1; 3; 6; 11]

b =
     -1
      3
      6
     11

b=[–1  3  6  11]'        % Observe the single quotation character (‘)

b =
     -1
      3
      6
     11
```

We will now define Matrix Multiplication and Element–by–Element multiplication.

1. **Matrix Multiplication** (multiplication of row by column vectors)

   Let

$$\mathbf{A} = [a_1 \quad a_2 \quad a_3 \quad \ldots \quad a_n]$$

   and

$$\mathbf{B} = [b_1 \quad b_2 \quad b_3 \quad \ldots \quad b_n]'$$

   be two vectors. We observe that **A** is defined as a row vector whereas **B** is defined as a column vector, as indicated by the transpose operator (′). Here, multiplication of the row vector **A** by the column vector **B**, is performed with the matrix multiplication operator (*). Then,

$$\mathbf{A*B} = [a_1b_1 + a_2b_2 + a_3b_3 + \ldots + a_nb_n] = \text{single value} \tag{A.5}$$

For example, if

$$\mathbf{A} = [1 \quad 2 \quad 3 \quad 4 \quad 5]$$

and

$$\mathbf{B} = [-2 \quad 6 \quad -3 \quad 8 \quad 7]'$$

the matrix multiplication **A*B** produces the single value 68, that is,

$$\mathbf{A*B} = 1 \times (-2) + 2 \times 6 + 3 \times (-3) + 4 \times 8 + 5 \times 7 = 68$$

and this is verified with the MATLAB script

```
A=[1  2  3  4  5]; B=[ –2  6  –3  8  7]'; A*B   % Observe transpose operator (') in B
ans =
  68
```

Now, let us suppose that both **A** and **B** are row vectors, and we attempt to perform a row–by–row multiplication with the following MATLAB statements.

```
A=[1  2  3  4  5]; B=[–2  6  –3  8  7]; A*B          % No transpose operator (') here
```

When these statements are executed, MATLAB displays the following message:

```
??? Error using ==> *
Inner matrix dimensions must agree.
```

Here, because we have used the matrix multiplication operator (*) in A*B, MATLAB expects vector **B** to be a column vector, not a row vector. It recognizes that **B** is a row vector, and warns us that we cannot perform this multiplication using the matrix multiplication operator (*). Accordingly, we must perform this type of multiplication with a different operator. This operator is defined below.

2. **Element–by–Element Multiplication** (multiplication of a row vector by another row vector)

Let

$$\mathbf{C} = [c_1 \quad c_2 \quad c_3 \quad \ldots \quad c_n]$$

and

$$\mathbf{D} = [d_1 \quad d_2 \quad d_3 \quad \ldots \quad d_n]$$

be two row vectors. Here, multiplication of the row vector **C** by the row vector **D** is performed with the ***dot multiplication operator*** (**.***). There is no space between the dot and the multiplication symbol. Thus,

$$\mathbf{C.*D} = [c_1d_1 \quad c_2d_2 \quad c_3d_3 \quad \ldots \quad c_nd_n] \tag{A.6}$$

This product is another row vector with the same number of elements, as the elements of **C** and **D**.

As an example, let

$$\mathbf{C} = [1 \quad 2 \quad 3 \quad 4 \quad 5]$$

and

$$\mathbf{D} = [-2 \quad 6 \quad -3 \quad 8 \quad 7]$$

Dot multiplication of these two row vectors produce the following result.

$$\mathbf{C.*D} = 1\times(-2) \quad 2\times 6 \quad 3\times(-3) \quad 4\times 8 \quad 5\times 7 = -2 \quad 12 \quad -9 \quad 32 \quad 35$$

Check with MATLAB:

```
C=[1  2   3   4  5];      %  Vectors C and D must have
D=[–2  6 –3   8  7];      %  same number of elements
C.*D                      % We observe that this is a dot multiplication
```

```
ans =
    -2     12     -9     32     35
```

Similarly, the division (/) and exponentiation (^) operators, are used for matrix division and exponentiation, whereas dot division (./) and dot exponentiation (.^) are used for element–by–element division and exponentiation, as illustrated in Examples A.11 and A.12 above.

We must remember that ***no space is allowed between the dot (.) and the multiplication, division, and exponentiation operators.***

Note: A dot (.) is never required with the plus (+) and minus (–) operators.

Example A.13

Write the MATLAB script that produces a simple plot for the waveform defined as

$$y = f(t) = 3e^{-4t}\cos 5t - 2e^{-3t}\sin 2t + \frac{t^2}{t+1} \tag{A.7}$$

in the $0 \le t \le 5$ seconds interval.

Solution:

The MATLAB script for this example is as follows:

```
t=0: 0.01: 5;                % Define t–axis in 0.01 increments
y=3 .* exp(–4 .* t) .* cos(5 .* t)–2 .* exp(–3 .* t) .* sin(2 .* t) + t .^2 ./ (t+1);
plot(t,y); grid; xlabel('t'); ylabel('y=f(t)'); title('Plot for Example A.13')
```

The plot for this example is shown in Figure A.8.

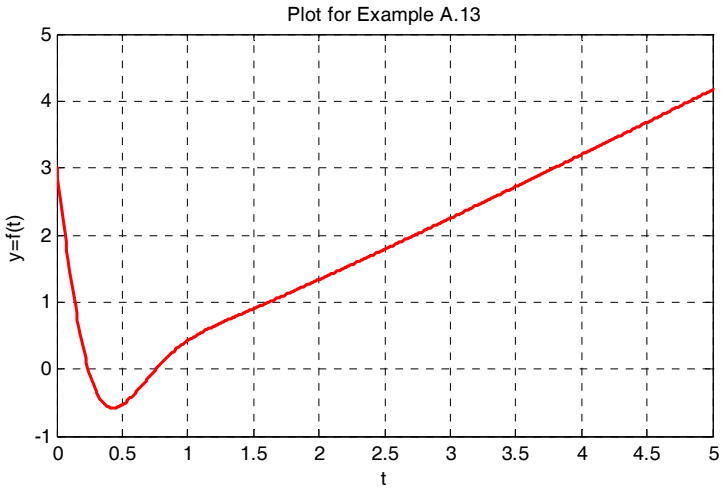


Figure A.8. Plot for Example A.13

Had we, in this example, defined the time interval starting with a negative value equal to or less than -1, say as $-3 \le t \le 3$, MATLAB would have displayed the following message:

```
Warning: Divide by zero.
```

This is because the last term (the rational fraction) of the given expression, is divided by zero when $t = -1$. To avoid division by zero, we use the special MATLAB function **eps,** which is a number approximately equal to 2.2×10^{-16}. It will be used with the next example.

The command **axis([xmin xmax ymin ymax])** scales the current plot to the values specified by the arguments **xmin, xmax, ymin and ymax.** There are no commas between these four arguments. This command must be placed ***after*** the plot command and must be repeated for each plot. The following example illustrates the use of the dot multiplication, division, and exponentiation, the **eps** number, the **axis([xmin xmax ymin ymax])** command, and also MATLAB's capability of displaying up to four windows of different plots.

Example A.14

Plot the functions

$$y = \sin^2 x, \quad z = \cos^2 x, \quad w = \sin^2 x \cdot \cos^2 x, \quad v = \sin^2 x / \cos^2 x$$

in the interval $0 \le x \le 2\pi$ using 100 data points. Use the **subplot** command to display these functions on four windows on the same graph.

Solution:

The MATLAB script to produce the four subplots is as follows:

```
x=linspace(0,2*pi,100);              % Interval with 100 data points
y=(sin(x).^ 2);  z=(cos(x).^ 2);
w=y.* z;
v=y./ (z+eps);%  add eps to avoid division by zero
subplot(221);% upper left of four subplots
plot(x,y);  axis([0 2*pi 0 1]);
title('y=(sinx) ^ 2');
subplot(222);                        % upper right of four subplots
plot(x,z);  axis([0 2*pi 0 1]);
title('z=(cosx) ^ 2');
subplot(223);                        % lower left of four subplots
plot(x,w);  axis([0 2*pi 0 0.3]);
title('w=(sinx) ^ 2*(cosx) ^ 2');
subplot(224);                        % lower right of four subplots
plot(x,v);  axis([0 2*pi 0 400]);
title('v=(sinx) ^ 2/(cosx) ^ 2');
```

These subplots are shown in Figure A.9.

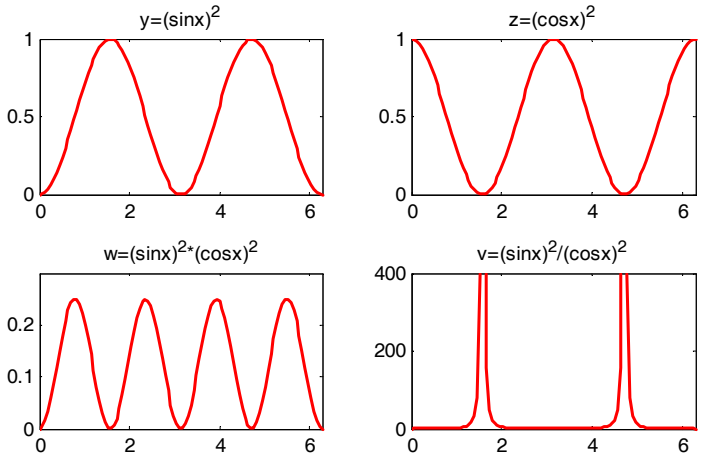


Figure A.9. Subplots for the functions of Example A.14

The next example illustrates MATLAB's capabilities with imaginary numbers. We will introduce the **real(z)** and **imag(z)** functions that display the real and imaginary parts of the complex quantity $z = x + iy$, the **abs(z)**, and the **angle(z)** functions that compute the absolute value (magnitude) and phase angle of the complex quantity $z = x + iy = r\angle\theta$. We will also use the **polar(theta,r)** function that produces a plot in polar coordinates, where **r** is the magnitude, **theta**

is the angle in radians, and the **round(n)** function that rounds a number to its nearest integer.

Example A.15

Consider the electric circuit of Figure A.10.

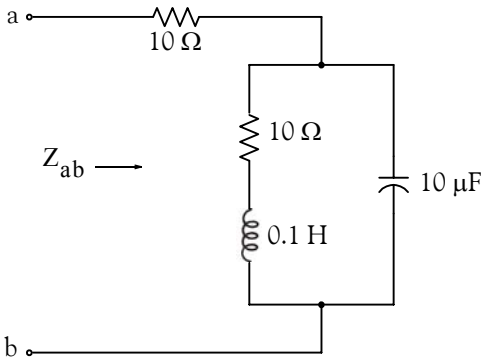


Figure A.10. Electric circuit for Example A.15

With the given values of resistance, inductance, and capacitance, the impedance Z_{ab} as a function of the radian frequency ω can be computed from the following expression:

$$Z_{ab} = Z = 10 + \frac{10^4 - j(10^6/\omega)}{10 + j(0.1\omega - 10^5/\omega)} \tag{A.8}$$

a. Plot Re{Z} (the real part of the impedance *Z*) versus frequency ω.

b. Plot Im{Z} (the imaginary part of the impedance *Z*) versus frequency ω.

c. Plot the impedance *Z* versus frequency ω in polar coordinates.

Solution:

The MATLAB script below computes the real and imaginary parts of Z_{ab} which, for simplicity, are denoted as **z**, and plots these as two separate graphs (parts a & b). It also produces a polar plot (part c).

```
w=0: 1: 2000;                    %  Define interval with one radian interval;...
z=(10+(10 .^ 4 –j .* 10 .^ 6 ./ (w+eps)) ./ (10 + j .* (0.1 .* w –10.^5./ (w+eps))));...
%
%  The first five statements (next two lines) compute and plot Re{z}
real_part=real(z);  plot(w,real_part);...
xlabel('radian frequency w');  ylabel('Real part of Z'); grid
```

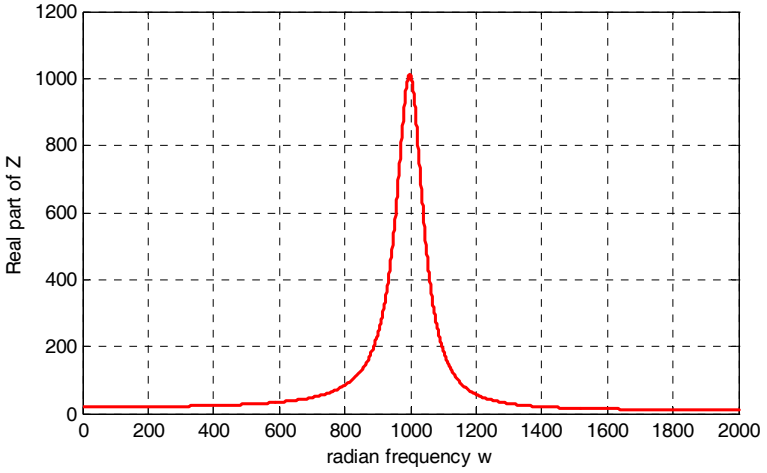


Figure A.11. Plot for the real part of the impedance in Example A.15

```
%  The next five statements (next two lines) compute and plot Im{z}
imag_part=imag(z);  plot(w,imag_part);...
xlabel('radian frequency w');  ylabel('Imaginary part of Z'); grid
```

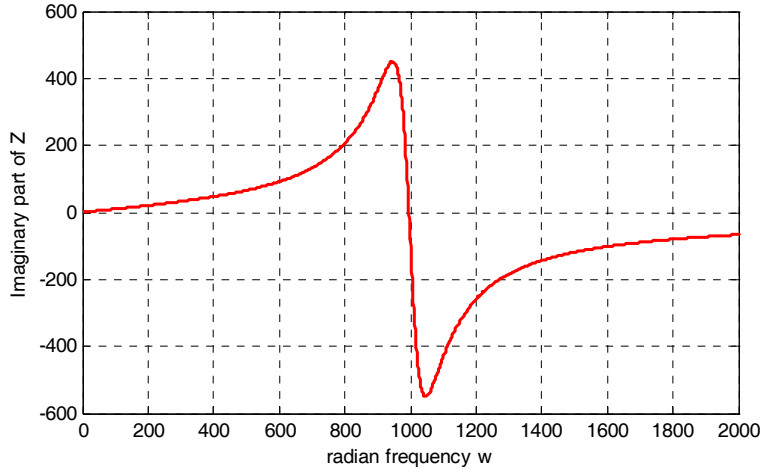


Figure A.12. Plot for the imaginary part of the impedance in Example A.15

```
%  The last six statements (next five lines) below produce the polar plot of z
mag=abs(z);                 %  Computes |Z|;...
rndz=round(abs(z));         %  Rounds |Z| to read polar plot easier;...
theta=angle(z);             %  Computes the phase angle of impedance Z;...
polar(theta,rndz);          %  Angle is the first argument
ylabel('Polar Plot of Z'); grid
```

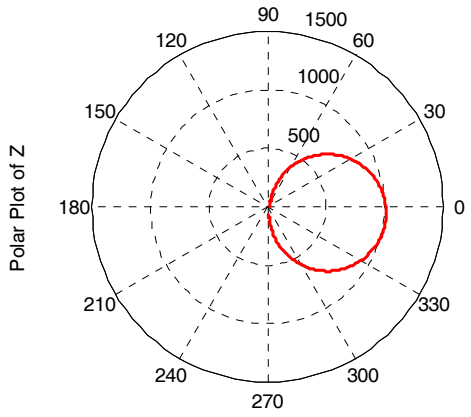


Figure A.13. Polar plot of the impedance in Example A.15

Example A.15 clearly illustrates how powerful, fast, accurate, and flexible MATLAB is.

A.10 Script and Function Files

MATLAB recognizes two types of files: ***script files*** and ***function files***. Both types are referred to as ***m–files*** since both require the *.m* extension.

A ***script file*** consists of two or more built–in functions such as those we have discussed thus far. Thus, the script for each of the examples we discussed earlier, make up a script file. Generally, a script file is one which was generated and saved as an m–file with an editor such as the MATLAB's Editor/Debugger.

A ***function file*** is a user–defined function using MATLAB. We use function files for repetitive tasks. The first line of a function file must contain the word *function*, followed by the output argument, the equal sign (=), and the input argument enclosed in parentheses. The function name and file name must be the same, but the file name must have the extension ***.m***. For example, the function file consisting of the two lines below

```
function y = myfunction(x)
y=x.^ 3 + cos(3.* x)
```

is a function file and must be saved as ***myfunction.m***

For the next example, we will use the following MATLAB functions:

fzero(f,x) – attempts to find a zero of a function of one variable, where **f** is a string containing the name of a real–valued function of a single real variable. MATLAB searches for a value near a point where the function **f** changes sign, and returns that value, or returns `NaN` if the search fails.

Important: We must remember that we use **roots(p)** to find the roots of polynomials only, such as those in Examples A.1 and A.2.

fplot(fcn,lims) – plots the function specified by the string **fcn** between the x–axis limits specified by **lims = [xmin xmax]**. Using **lims = [xmin xmax ymin ymax]** also controls the y–axis limits. The string **fcn** must be the name of an *m–file* function or a string with variable x .

NaN (Not–a–Number) is not a function; it is MATLAB's response to an undefined expression such as $0/0$, ∞/∞, or inability to produce a result as described on the next paragraph. We can avoid division by zero using the **eps** number, which we mentioned earlier.

Example A.16

Find the zeros, the minimum, and the maximum values of the function

$$f(x) = \frac{1}{(x-0.1)^2 + 0.01} - \frac{1}{(x-1.2)^2 + 0.04} - 10 \tag{A.9}$$

in the interval $-1.5 \le x \le 1.5$

Solution:

We first plot this function to observe the approximate zeros, maxima, and minima using the following script.

```
x=–1.5: 0.01: 1.5;
y=1./ ((x–0.1).^ 2 + 0.01) –1./ ((x–1.2).^ 2 + 0.04) –10;
plot(x,y); grid
```

The plot is shown in Figure A.14.

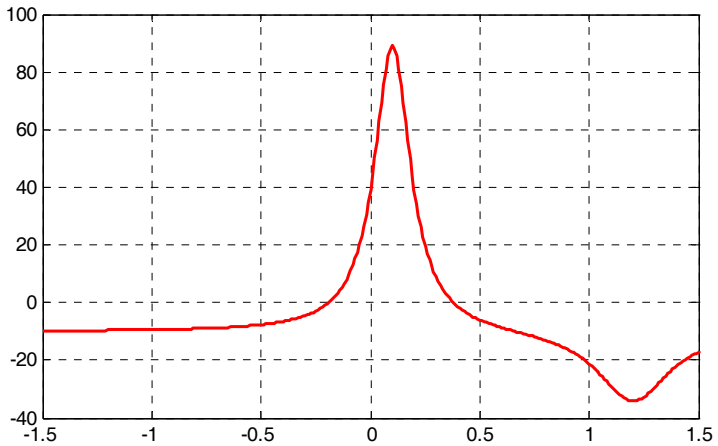


Figure A.14. Plot for Example A.16 using the plot command

The roots (zeros) of this function appear to be in the neighborhood of $x = -0.2$ and $x = 0.3$. The maximum occurs at approximately $x = 0.1$ where, approximately, $y_{max} = 90$, and the minimum occurs at approximately $x = 1.2$ where, approximately, $y_{min} = -34$.

Next, we define and save *f(x)* as the **funczero01.m** function m–file with the following script:

```
function y=funczero01(x)
% Finding the zeros of the function shown below
y=1/((x–0.1)^2+0.01)–1/((x–1.2)^2+0.04)–10;
```

To save this file, from the File drop menu on the Command Window, we choose New, and when the Editor Window appears, we type the script above and we save it as **funczero01.** MATLAB appends the extension **.m** to it.

Now, we can use the **fplot(fcn,lims)** command to plot f(x) as follows:

```
fplot('funczero01', [–1.5  1.5]); grid
```

This plot is shown in Figure A.15. As expected, this plot is identical to the plot of Figure A.14 which was obtained with the **plot(x,y)** command as shown in Figure A.14.

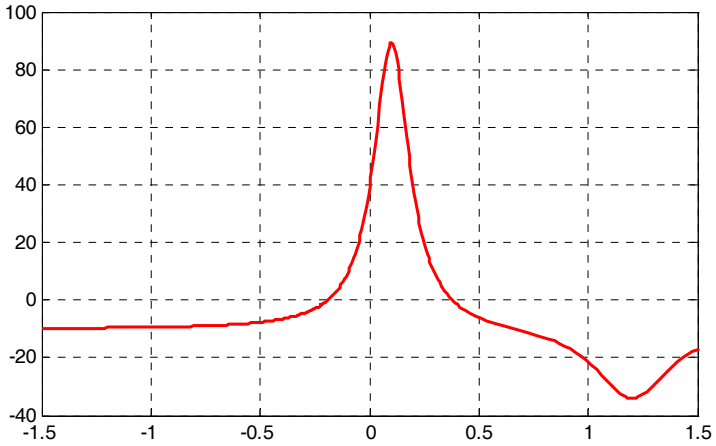


Figure A.15. Plot for Example A.16 using the fplot command

We will use the **fzero(f,x)** function to compute the roots of f(x) in Equation (A.9) more precisely. The MATLAB script below will accomplish this.

```
x1= fzero('funczero01', –0.2);
x2= fzero('funczero01', 0.3);
fprintf('The roots (zeros) of this function are r1= %3.4f', x1);
fprintf(' and r2= %3.4f \n', x2)
```

MATLAB displays the following:

```
The roots (zeros) of this function are r1= -0.1919 and r2= 0.3788
```

The earlier MATLAB versions included the function **fmin(f,x1,x2)** and with this function we could compute both a minimum of some function f(x) or a maximum of f(x) since a maximum of f(x) is equal to a minimum of –f(x). This can be visualized by flipping the plot of a function f(x) upside–down. This function is no longer used in MATLAB and thus we will compute the maxima and minima from the derivative of the given function.

From elementary calculus, we recall that the maxima or minima of a function y = f(x) can be found by setting the first derivative of a function equal to zero and solving for the independent variable x. For this example we use the **diff(x)** function which produces the approximate derivative of a function. Thus, we use the following MATLAB script:

```
syms x ymin zmin; ymin=1/((x–0.1)^2+0.01)–1/((x–1.2)^2+0.04)–10;...
zmin=diff(ymin)
```

```
zmin =
-1/((x-1/10)^2+1/100)^2*(2*x-1/5)+1/((x-6/5)^2+1/25)^2*(2*x-12/5)
```

When the command

```
solve(zmin)
```

is executed, MATLAB displays a very long expression which when copied at the command prompt and executed, produces the following:

```
ans =
   0.6585 + 0.3437i
ans =
   0.6585 - 0.3437i
ans =
    1.2012
```

The real value 1.2012 above is the value of x at which the function y has its minimum value as we observe also in the plot of Figure A.15.

To find the value of y corresponding to this value of *x*, we substitute it into f(x), that is,

```
x=1.2012; ymin=1 / ((x–0.1) ^ 2 + 0.01) –1 / ((x–1.2) ^ 2 + 0.04) –10
```

```
ymin = -34.1812
```

We can find the maximum value from –f(x) whose plot is produced with the script

```
x=–1.5:0.01:1.5; ymax=–1./((x–0.1).^2+0.01)+1./((x–1.2).^2+0.04)+10; plot(x,ymax); grid
```

and the plot is shown in Figure A.16.

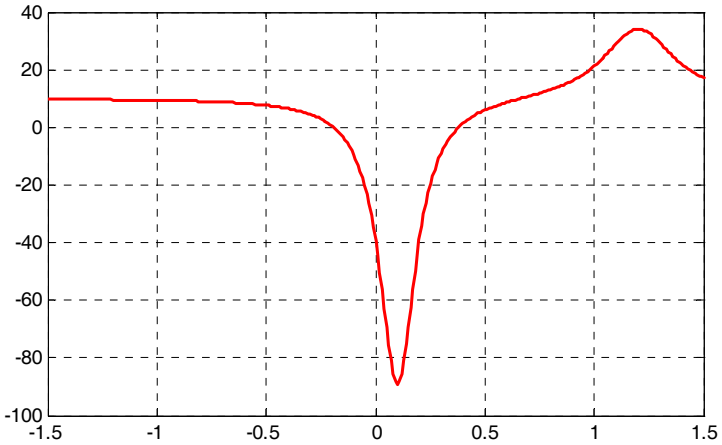


Figure A.16. Plot of –f(x) *for Example A.16*

Next we compute the first derivative of –f(x) and we solve for x to find the value where the maximum of ymax occurs. This is accomplished with the MATLAB script below.

```
syms x ymax zmax; ymax=–(1/((x–0.1)^2+0.01)–1/((x–1.2)^2+0.04)–10); zmax=diff(ymax)
```

```
zmax =
 1/((x-1/10)^2+1/100)^2*(2*x-1/5)-1/((x-6/5)^2+1/25)^2*(2*x-12/5)
```

```
solve(zmax)
```

When the command

```
solve(zmax)
```

is executed, MATLAB displays a very long expression which when copied at the command prompt and executed, produces the following:

```
ans =
   0.6585 + 0.3437i

ans =
   0.6585 - 0.3437i

ans =
    1.2012
ans =
    0.0999
```

From the values above we choose x = 0.0999 which is consistent with the plots of Figures A.15 and A.16. Accordingly, we execute the following script to obtain the value of ymin.

```
x=0.0999;                    % Using this value find the corresponding value of ymax
ymax=1 / ((x–0.1) ^ 2 + 0.01) –1 / ((x–1.2) ^ 2 + 0.04) –10

ymax = 89.2000
```

A.11 Display Formats

MATLAB displays the results on the screen in integer format without decimals if the result is an integer number, or in short floating point format with four decimals if it a fractional number. The format displayed has nothing to do with the accuracy in the computations. MATLAB performs all computations with accuracy up to 16 decimal places.

The output format can changed with the **format** command. The available MATLAB formats can be displayed with the **help format** command as follows:

```
help format

FORMAT Set output format.
All computations in MATLAB are done in double precision.
FORMAT may be used to switch between different output display formats
as follows:

FORMAT  Default. Same as SHORT.
FORMAT SHORT Scaled fixed point format with 5 digits.
FORMAT LONG Scaled fixed point format with 15 digits.
FORMAT SHORT E Floating point format with 5 digits.
FORMAT LONG E  Floating point format with 15 digits.
FORMAT SHORT G Best of fixed or floating point format with 5 digits.
FORMAT LONG G Best of fixed or floating point format with 15 digits.
FORMAT HEX Hexadecimal format.
FORMAT + The symbols +, - and blank are printed for positive, negative,
         and zero elements.Imaginary parts are ignored.
FORMAT BANK Fixed format for dollars and cents.
FORMAT RAT Approximation by ratio of small integers.

Spacing:

FORMAT COMPACT Suppress extra line-feeds.
FORMAT LOOSE  Puts the extra line-feeds back in.

Some examples with different format displays age given below.

format short  33.3335  Four decimal digits (default)
format long  33.33333333333334 16 digits
format short e  3.3333e+01  Four decimal digits plus exponent
format short g  33.333  Better of format short or format short e
format bank  33.33 two decimal digits
format +  only + or - or zero are printed
```

```
format rat 100/3 rational approximation
```

The **disp(X)** command displays the array **X** without printing the array name. If **X** is a string, the text is displayed.

The **fprintf(format,array)** command displays and prints both text and arrays. It uses specifiers to indicate where and in which format the values would be displayed and printed. Thus, if **%f** is used, the values will be displayed and printed in fixed decimal format, and if **%e** is used, the values will be displayed and printed in scientific notation format. With this command only the real part of each parameter is processed.

This appendix is just an introduction to MATLAB.* This outstanding software package consists of many applications known as *Toolboxes*. The MATLAB Student Version contains just a few of these Toolboxes. Others can be bought directly from The MathWorks,™ Inc., as add–ons.

* *For more MATLAB applications, please refer to Numerical Analysis Using MATLAB and Spreadsheets, ISBN 0–9709511–1–6.*

Appendix B

Introduction to Simulink®

This appendix is a brief introduction to Simulink. This author feels that we can best introduce Simulink with a few examples. Some familiarity with MATLAB is essential in understanding Simulink, and for this purpose, Appendix A is included as an introduction to MATLAB.

B.1 Simulink and its Relation to MATLAB

The MATLAB® and Simulink® environments are integrated into one entity, and thus we can analyze, simulate, and revise our models in either environment at any point. We invoke Simulink from within MATLAB. We will introduce Simulink with a few illustrated examples.

Example B.1

For the circuit of Figure B.1, the initial conditions are $i_L(0^-) = 0$, and $v_c(0^-) = 0.5$ V. We will compute $v_c(t)$.

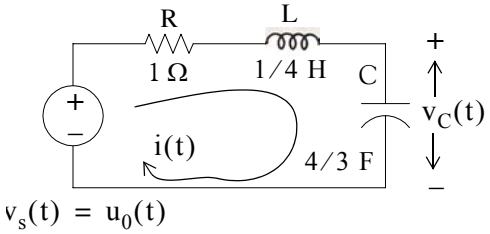


Figure B.1. Circuit for Example B.1

For this example,

$$i = i_L = i_C = C\frac{dv_C}{dt} \tag{B.1}$$

and by Kirchoff's voltage law (KVL),

$$Ri_L + L\frac{di_L}{dt} + v_C = u_0(t) \tag{B.2}$$

Substitution of (B.1) into (B.2) yields

$$RC\frac{dv_C}{dt} + LC\frac{d^2v_C}{dt^2} + v_C = u_0(t) \tag{B.3}$$

Substituting the values of the circuit constants and rearranging we obtain:

$$\frac{1}{3}\frac{d^2v_C}{dt^2} + \frac{4}{3}\frac{dv_C}{dt} + v_C = u_0(t)$$

$$\frac{d^2v_C}{dt^2} + 4\frac{dv_C}{dt} + 3v_C = 3u_0(t) \tag{B.4}$$

$$\frac{d^2v_C}{dt^2} + 4\frac{dv_C}{dt} + 3v_C = 3 \qquad t > 0 \tag{B.5}$$

To appreciate Simulink's capabilities, for comparison, three different methods of obtaining the solution are presented, and the solution using Simulink follows.

First Method – Assumed Solution

Equation (B.5) is a second–order, non–homogeneous differential equation with constant coefficients, and thus the complete solution will consist of the sum of the forced response and the natural response. It is obvious that the solution of this equation cannot be a constant since the derivatives of a constant are zero and thus the equation is not satisfied. Also, the solution cannot contain sinusoidal functions (sine and cosine) since the derivatives of these are also sinusoids. However, decaying exponentials of the form ke^{-at} where k and a are constants, are possible candidates since their derivatives have the same form but alternate in sign.

It can be shown* that if $k_1e^{-s_1t}$ and $k_2e^{-s_2t}$ where k_1 and k_2 are constants and s_1 and s_2 are the roots of the characteristic equation of the homogeneous part of the given differential equation, the natural response is the sum of the terms $k_1e^{-s_1t}$ and $k_2e^{-s_2t}$. Therefore, the total solution will be

$$v_c(t) = \text{natural response} + \text{forced response} = v_{cn}(t) + v_{cf}(t) = k_1e^{-s_1t} + k_2e^{-s_2t} + v_{cf}(t) \tag{B.6}$$

The values of s_1 and s_2 are the roots of the characteristic equation

* *Please refer to Circuit Analysis II with MATLAB Applications, ISBN 0–9709511–5–9, Appendix B for a thorough discussion.*

$$s^2 + 4s + 3 = 0 \tag{B.7}$$

Solution of (B.7) yields of $s_1 = -1$ and $s_2 = -3$ and with these values (B.6) is written as

$$v_c(t) = k_1 e^{-t} + k_2 e^{-3t} + v_{cf}(t) \tag{B.8}$$

The forced component $v_{cf}(t)$ is found from (B.5), i.e.,

$$\frac{d^2 v_C}{dt^2} + 4\frac{dv_C}{dt} + 3v_C = 3 \qquad t > 0 \tag{B.9}$$

Since the right side of (B.9) is a constant, the forced response will also be a constant and we denote it as $v_{Cf} = k_3$. By substitution into (B.9) we obtain

$$0 + 0 + 3k_3 = 3$$

or

$$v_{Cf} = k_3 = 1 \tag{B.10}$$

Substitution of this value into (B.8), yields the total solution as

$$v_C(t) = v_{Cn}(t) + v_{Cf} = k_1 e^{-t} + k_2 e^{-3t} + 1 \tag{B.11}$$

The constants k_1 and k_2 will be evaluated from the initial conditions. First, using $v_C(0) = 0.5$ V and evaluating (B.11) at $t = 0$, we obtain

$$v_C(0) = k_1 e^0 + k_2 e^0 + 1 = 0.5$$

$$k_1 + k_2 = -0.5 \tag{B.12}$$

Also,

$$i_L = i_C = C\frac{dv_C}{dt}, \quad \frac{dv_C}{dt} = \frac{i_L}{C}$$

and

$$\left.\frac{dv_C}{dt}\right|_{t=0} = \frac{i_L(0)}{C} = \frac{0}{C} = 0 \tag{B.13}$$

Next, we differentiate (B.11), we evaluate it at $t = 0$, and equate it with (B.13). Thus,

$$\left.\frac{dv_C}{dt}\right|_{t=0} = -k_1 - 3k_2 \tag{B.14}$$

By equating the right sides of (B.13) and (B.14) we obtain

$$-k_1 - 3k_2 = 0 \tag{B.15}$$

Simultaneous solution of (B.12) and (B.15), gives $k_1 = -0.75$ and $k_2 = 0.25$. By substitution into (B.8), we obtain the total solution as

$$\mathbf{v_C(t) = (-0.75e^{-t} + 0.25e^{-3t} + 1)u_0(t)} \tag{B.16}$$

Check with MATLAB:

```
syms t                                          % Define symbolic variable t
y0=-0.75*exp(-t)+0.25*exp(-3*t)+1;              % The total solution y(t), for our example, vc(t)
y1=diff(y0)                                     % The first derivative of y(t)

y1 =
3/4*exp(-t)-3/4*exp(-3*t)

y2=diff(y0,2)                                   % The second derivative of y(t)

y2 =
-3/4*exp(-t)+9/4*exp(-3*t)

y=y2+4*y1+3*y0                                  % Summation of y and its derivatives

y =
3
```

Thus, the solution has been verified by MATLAB. Using the expression for $v_C(t)$ in (B.16), we find the expression for the current as

$$\mathbf{i = i_L = i_C = C\frac{dv_C}{dt} = \frac{4}{3}\left(\frac{3}{4}e^{-t} - \frac{3}{4}e^{-3t}\right) = e^{-t} - e^{-3t}\ A} \tag{B.17}$$

Second Method – Using the Laplace Transformation

The transformed circuit is shown in Figure B.2.

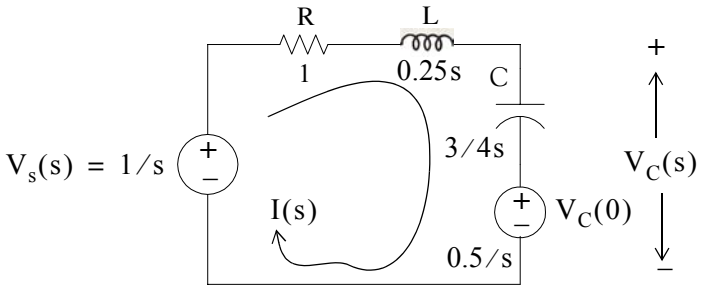


Figure B.2. Transformed Circuit for Example B.1

By the voltage division* expression,

$$V_C(s) = \frac{3/4s}{(1 + 0.25s + 3/4s)} \cdot \left(\frac{1}{s} - \frac{0.5}{s}\right) + \frac{0.5}{s} = \frac{1.5}{s(s^2 + 4s + 3)} + \frac{0.5}{s} = \frac{0.5s^2 + 2s + 3}{s(s+1)(s+3)}$$

Using partial fraction expansion,† we let

$$\frac{0.5s^2 + 2s + 3}{s(s+1)(s+3)} = \frac{r_1}{s} + \frac{r_2}{(s+1)} + \frac{r_3}{(s+3)} \tag{B.18}$$

$$r_1 = \left.\frac{0.5s^2 + 2s + 3}{(s+1)(s+3)}\right|_{s=0} = 1$$

$$r_2 = \left.\frac{0.5s^2 + 2s + 3}{s(s+3)}\right|_{s=-1} = -0.75$$

$$r_3 = \left.\frac{0.5s^2 + 2s + 3}{s(s+1)}\right|_{s=-3} = 0.25$$

and by substitution into (B.18)

$$V_C(s) = \frac{0.5s^2 + 2s + 3}{s(s+1)(s+3)} = \frac{1}{s} + \frac{-0.75}{(s+1)} + \frac{0.25}{(s+3)}$$

Taking the Inverse Laplace transform‡ we find that

$$\mathbf{v_C(t) = 1 - 0.75e^{-t} + 0.25e^{-3t}}$$

Third Method – Using State Variables

$$Ri_L + L\frac{di_L}{dt} + v_C = u_0(t)^{**}$$

---

* *For derivation of the voltage division and current division expressions, please refer to Circuit Analysis I with MATLAB Applications, ISBN 0–9709511–2–4.*

† *Partial fraction expansion is discussed in Chapter 3, this text.*

‡ *For an introduction to Laplace Transform and Inverse Laplace Transform, please refer Chapters 2 and 3, this text.*

** *Usually, in State–Space and State Variables Analysis,* $u(t)$ *denotes any input. For distinction, we will denote the Unit Step Function as* $u_0(t)$. *For a detailed discussion on State–Space and State Variables Analysis, please refer to Chapter 5, this text.*

By substitution of given values and rearranging, we obtain

$$\frac{1}{4}\frac{di_L}{dt} = (-1)i_L - v_C + 1$$

or

$$\frac{di_L}{dt} = -4i_L - 4v_C + 4 \tag{B.19}$$

Next, we define the state variables $x_1 = i_L$ and $x_2 = v_C$. Then,

$$\dot{x}_1 = \frac{di_L}{dt}^* \tag{B.20}$$

and

$$\dot{x}_2 = \frac{dv_C}{dt} \tag{B.21}$$

Also,

$$i_L = C\frac{dv_C}{dt}$$

and thus,

$$x_1 = i_L = C\frac{dv_C}{dt} = C\dot{x}_2 = \frac{4}{3}\dot{x}_2$$

or

$$\dot{x}_2 = \frac{3}{4}x_1 \tag{B.22}$$

Therefore, from (B.19), (B.20), and (B.22), we obtain the state equations

$$\dot{x}_1 = -4x_1 - 4x_2 + 4$$

$$\dot{x}_2 = \frac{3}{4}x_1$$

and in matrix form,

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-4 & -4\\ 3/4 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}4\\ 0\end{bmatrix}u_0(t) \tag{B.23}$$

Solution† of (B.23) yields

---

* *The notation* $\dot{x}$ *(x dot) is often used to denote the first derivative of the function* x*, that is,* $\dot{x} = dx/dt$.

† *The detailed solution of (B.23) is given in Chapter 5, Example 5.10, Page 5–23, this text.*

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} e^{-t} - e^{-3t} \\ 1 - 0.75e^{-t} + 0.25e^{-3t} \end{bmatrix}$$

Then,

$$x_1 = i_L = e^{-t} - e^{-3t} \tag{B.24}$$

and

$$x_2 = v_C = 1 - 0.75e^{-t} + 0.25e^{-3t} \tag{B.25}$$

Modeling the Differential Equation of Example B.1 with Simulink

To run Simulink, we must first invoke MATLAB. Make sure that Simulink is installed in your system. In the MATLAB Command prompt, we type:

```
simulink
```

Alternately, we can click on the Simulink icon shown in Figure B.3. It appears on the top bar on MATLAB's Command prompt.

Figure B.3. The Simulink icon

Upon execution of the Simulink command, the **Commonly Used Blocks** appear as shown in Figure B.4.

In Figure B.4, the left side is referred to as the **Tree Pane** and displays all Simulink libraries installed. The right side is referred to as the **Contents Pane** and displays the blocks that reside in the library currently selected in the Tree Pane.

Let us express the differential equation of Example B.1 as

$$\frac{d^2 v_C}{dt^2} = -4\frac{dv_C}{dt} - 3v_C + 3u_0(t) \tag{B.26}$$

A block diagram representing relation (B.26) above is shown in Figure B.5. We will use Simulink to draw a similar block diagram.*

* *Henceforth, all Simulink block diagrams will be referred to as models.*

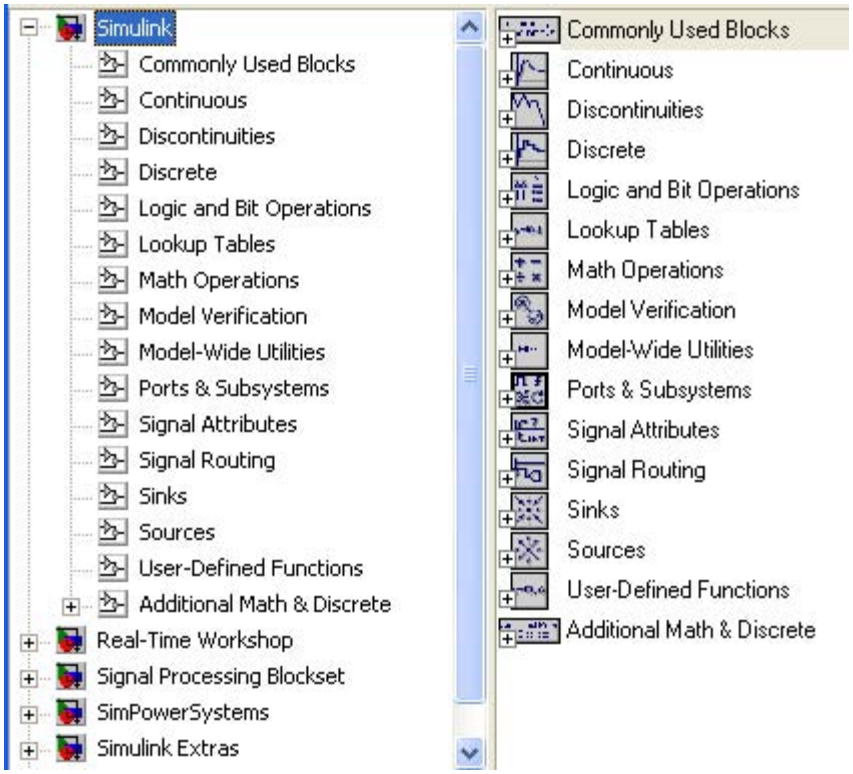


Figure B.4. The Simulink Library Browser

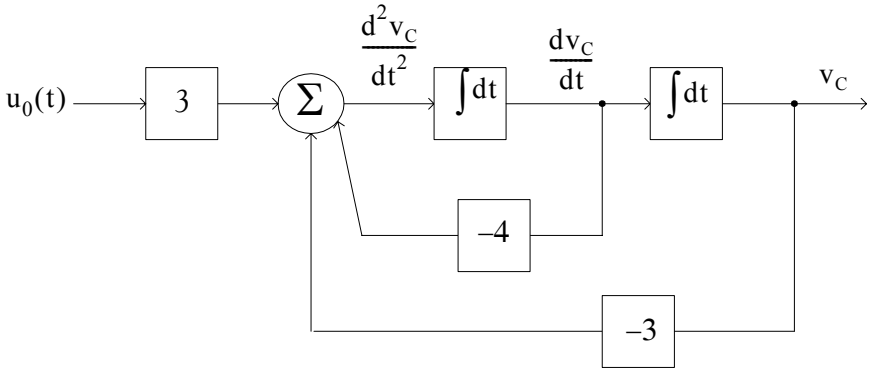


Figure B.5. Block diagram for equation (B.26)

To model the differential equation (B.26) using Simulink, we perform the following steps:

1. On the **Simulink Library Browser**, we click on the leftmost icon shown as a blank page on the top title bar. A new model window named **untitled** will appear as shown in Figure B.6.

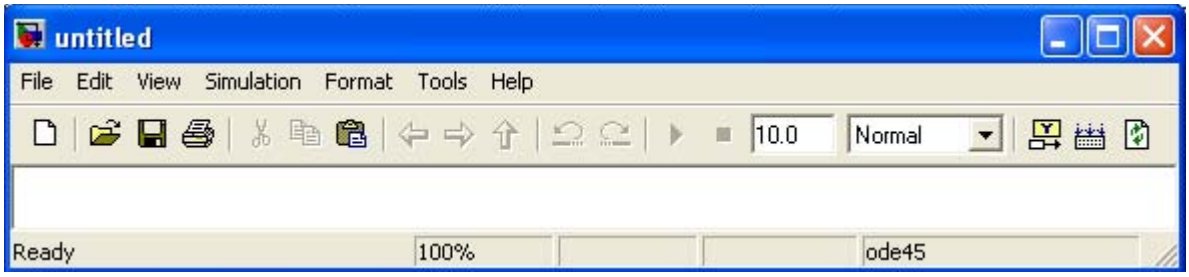


Figure B.6. The Untitled model window in Simulink.

The window of Figure B.6 is the model window where we enter our blocks to form a block diagram. We save this as model file name Equation_1_26. This is done from the File drop menu of Figure B.6 where we choose **Save as** and name the file as Equation_1_26. Simulink will add the extension .mdl. The new model window will now be shown as **Equation_1_26**, and all saved files will have this appearance. See Figure B.7.

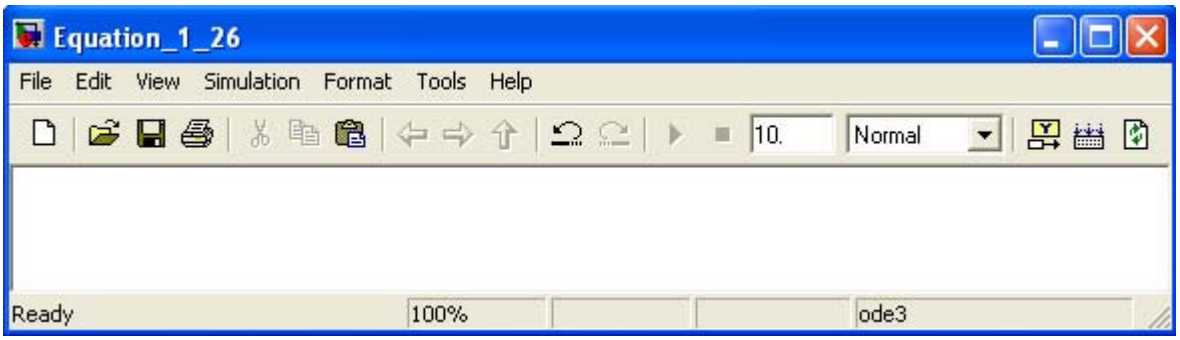


Figure B.7. Model window for Equation_1_26.mdl file

2. With the **Equation_1_26** model window and the **Simulink Library Browser** both visible, we click on the **Sources** appearing on the left side list, and on the right side we scroll down until we see the unit step function shown as **Step**. See Figure B.8. We select it, and we drag it into the **Equation_1_26** model window which now appears as shown in Figure B.8. We save file Equation_1_26 using the File drop menu on the **Equation_1_26** model window (right side of Figure B.8).

3. With reference to block diagram of Figure B.5, we observe that we need to connect an amplifier with Gain 3 to the unit step function block. The gain block in Simulink is under **Commonly Used Blocks** (first item under Simulink on the **Simulink Library Browser**). See Figure B.8. If the **Equation_1_26** model window is no longer visible, it can be recalled by clicking on the white page icon on the top bar of the **Simulink Library Browser**.

4. We choose the gain block and we drag it to the right of the unit step function. The triangle on the right side of the unit step function block and the > symbols on the left and right sides of the gain block are connection points. We point the mouse close to the connection point of the unit step function until is shows as a cross hair, and draw a straight line to connect the two

blocks.* We double–click on the gain block and on the **Function Block Parameters**, we change the gain from 1 to 3. See Figure B.9.

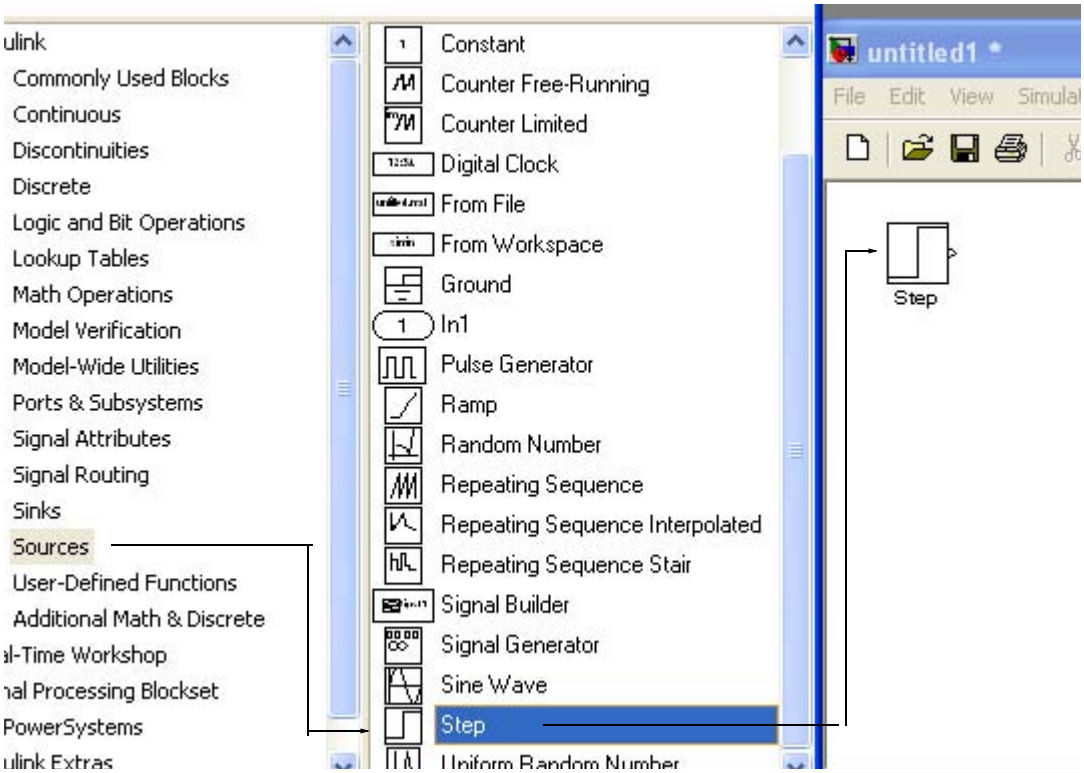


Figure B.8. Dragging the unit step function into File Equation_1_26

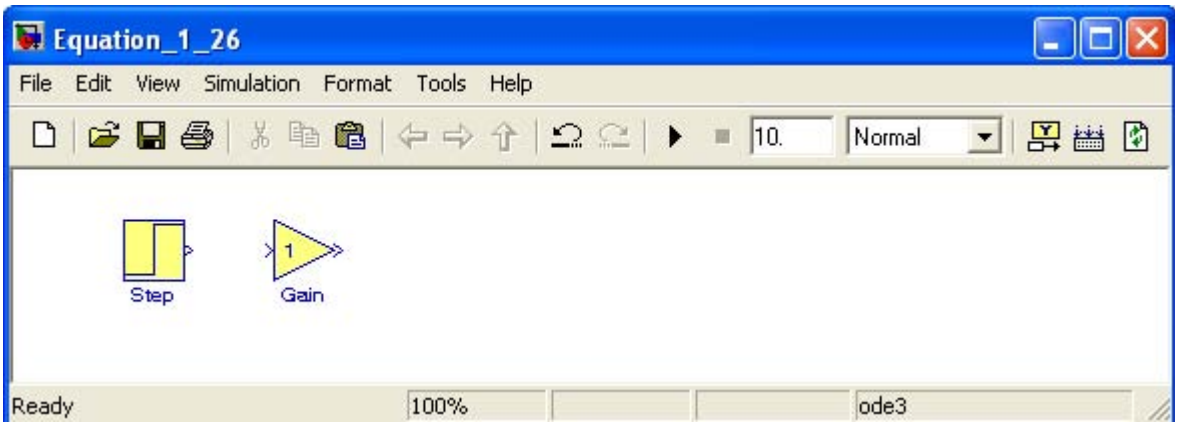


Figure B.9. File Equation_1_26 with added Step and Gain blocks

* *An easy method to interconnect two Simulink blocks by clicking on the source block to select it, then hold down the* **Ctrl** *key and left–click on the destination block.*

5. Next, we need to add a thee–input adder. The adder block appears on the right side of the **Simulink Library Browser** under **Math Operations**. We select it, and we drag it into the Equation_1_26 model window. We double click it, and on the **Function Block Parameters** window which appears**,** we specify 3 inputs. We then connect the output of the of the gain block to the first input of the adder block as shown in Figure B.10.

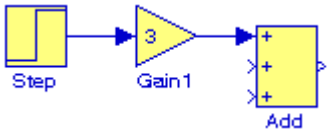


Figure B.10. File Equation_1_26 with added gain block

6. From the **Commonly Used Blocks** of the **Simulink Library Browser**, we choose the **Integrator** block, we drag it into the Equation_1_26 model window, and we connect it to the output of the **Add** block. We repeat this step and to add a second **Integrator** block. We click on the text "Integrator" under the first integrator block, and we change it to Integrator 1. Then, we change the text "Integrator 1" under the second Integrator to "Integrator 2" as shown in Figure B.11.

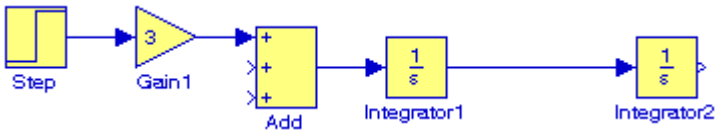


Figure B.11. File Equation_1_26 with the addition of two integrators

7. To complete the block diagram, we add the **Scope** block which is found in the **Commonly Used Blocks** on the **Simulink Library Browser**, we click on the Gain block, and we copy and paste it twice. We flip the pasted Gain blocks by using the **Flip Block** command from the Format drop menu, and we label these as Gain 2 and Gain 3. Finally, we double–click on these gain blocks and on the **Function Block Parameters** window, we change the gains from to –4 and –3 as shown in Figure B.12.

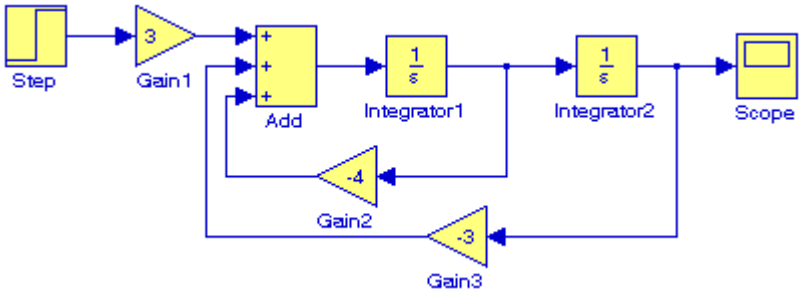


Figure B.12. File Equation_1_26 complete block diagram

8. The initial conditions $i_L(0^-) = C(dv_C/dt)\big|_{t=0} = 0$, and $v_c(0^-) = 0.5$ V are entered by double clicking the Integrator blocks and entering the values 0 for the first integrator, and 0.5 for the second integrator. We also need to specify the simulation time. This is done by specifying the simulation time to be 10 seconds on the **Configuration Parameters** from the **Simulation** drop menu. We can start the simulation on **Start** from the **Simulation** drop menu or by clicking on the  icon.

9. To see the output waveform, we double click on the **Scope** block, and then clicking on the Autoscale  icon, we obtain the waveform shown in Figure B.13.

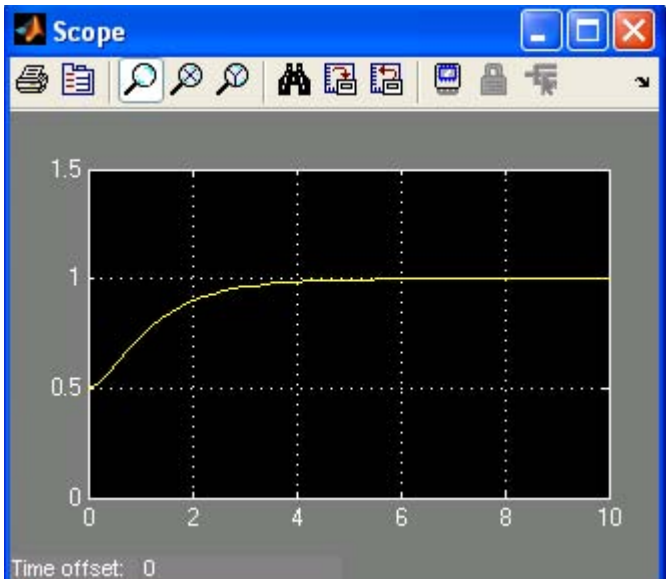


Figure B.13. The waveform for the function $v_C(t)$ for Example B.1

Another easier method to obtain and display the output $v_C(t)$ for Example B.1, is to use **State–Space** block from **Continuous** in the Simulink Library Browser, as shown in Figure B.14.

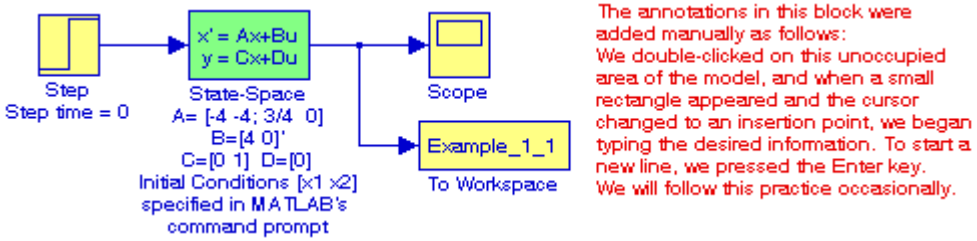


Figure B.14. Obtaining the function $v_C(t)$ for Example B.1 with the State–Space block.

The **simout To Workspace** block shown in Figure B.14 writes its input to the workspace. The data and variables created in the MATLAB Command window, reside in the MATLAB Workspace. This block writes its output to an array or structure that has the name specified by the block's Variable name parameter. This gives us the ability to delete or modify selected variables. We issue the command who to see those variables. From Equation B.23, Page B–6,

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-4 & -4\\ 3/4 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}4\\ 0\end{bmatrix}u_0(t)$$

The output equation is

$$y = Cx + du$$

or

$$y = [0 \quad 1]\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]u$$

We double–click on the **State–Space** block, and in the **Functions Block Parameters** window we enter the constants shown in Figure B.15.

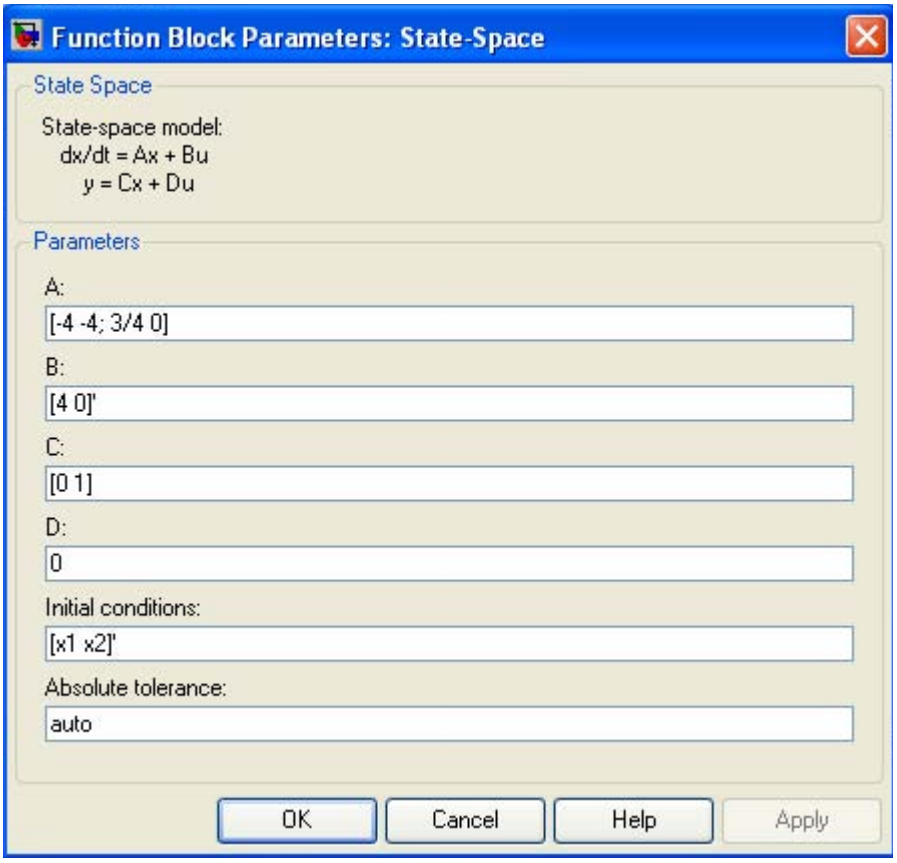


Figure B.15. The Function block parameters for the State–Space block.

The initials conditions [x1 x2]' are specified in MATLAB's Command prompt as

```
x1=0; x2=0.5;
```

As before, to start the simulation we click clicking on the  icon, and to see the output waveform, we double click on the **Scope** block, and then clicking on the Autoscale  icon, we obtain the waveform shown in Figure B.16.

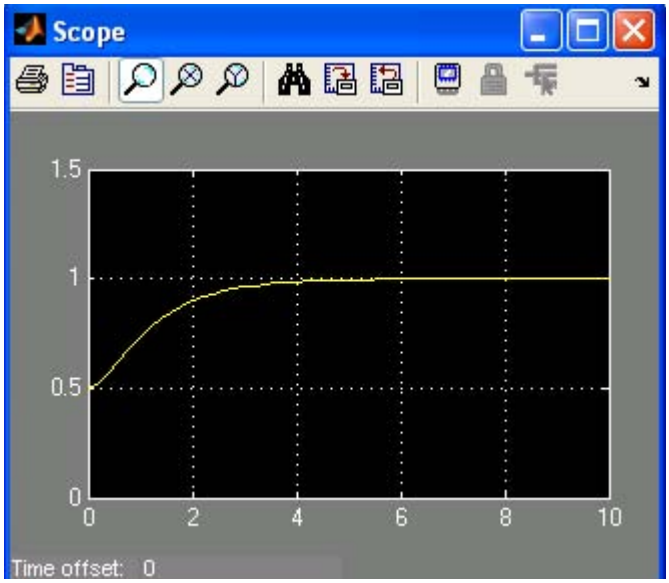


Figure B.16. The waveform for the function $v_C(t)$ for Example B.1 with the State–Space block.

The state–space block is the best choice when we need to display the output waveform of three or more variables as illustrated by the following example.

Example B.2

A fourth–order network is described by the differential equation

$$\frac{d^4y}{dt^4} + a_3\frac{d^3y}{dt^3} + a_2\frac{d^2y}{dt^2} + a_1\frac{dy}{dt} + a_0\, y(t) = u(t) \tag{B.27}$$

where $y(t)$ is the output representing the voltage or current of the network, and $u(t)$ is any input, and the initial conditions are $y(0) = y'(0) = y''(0) = y'''(0) = 0$.

a. We will express (B.27) as a set of state equations

b. It is known that the solution of the differential equation

$$\frac{d^4y}{dt^4} + 2\frac{d^2y}{dt^2} + y(t) = \sin t \tag{B.28}$$

subject to the initial conditions $y(0) = y'(0) = y''(0) = y'''(0) = 0$, has the solution

$$y(t) = 0.125[(3 - t^2) - 3t\cos t] \tag{B.29}$$

In our set of state equations, we will select appropriate values for the coefficients $a_3, a_2, a_1,$ and a_0 so that the new set of the state equations will represent the differential equation of (B.28), and using Simulink, we will display the waveform of the output $y(t)$.

1. The differential equation of (B.28) is of fourth–order; therefore, we must define four state variables that will be used with the four first–order state equations.

We denote the state variables as x_1, x_2, x_3, and x_4, and we relate them to the terms of the given differential equation as

$$x_1 = y(t) \qquad x_2 = \frac{dy}{dt} \qquad x_3 = \frac{d^2y}{dt^2} \qquad x_4 = \frac{d^3y}{dt^3} \tag{B.30}$$

We observe that

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= x_4 \\ \frac{d^4y}{dt^4} &= \dot{x}_4 = -a_0x_1 - a_1x_2 - a_2x_3 - a_3x_4 + u(t) \end{aligned} \tag{B.31}$$

and in matrix form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t) \tag{B.32}$$

In compact form, (B.32) is written as

$$\dot{x} = Ax + bu \tag{B.33}$$

Also, the output is

$$y = Cx + du \tag{B.34}$$

where

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \text{and } u = u(t) \tag{B.35}$$

and since the output is defined as

$$y(t) = x_1$$

relation (B.34) is expressed as

$$y = [1 \ 0 \ 0 \ 0] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0]u(t) \tag{B.36}$$

2. By inspection, the differential equation of (B.27) will be reduced to the differential equation of (B.28) if we let

$$a_3 = 0 \qquad a_2 = 2 \qquad a_1 = 0 \qquad a_0 = 1 \qquad u(t) = \sin t$$

and thus the differential equation of (B.28) can be expressed in state–space form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & 0 & -2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sin t \tag{B.37}$$

where

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & 0 & -2 & 0 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \text{and } u = \sin t \tag{B.38}$$

Since the output is defined as

$$y(t) = x_1$$

in matrix form it is expressed as

$$y = [1\ \ 0\ \ 0\ \ 0] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0]\sin t \tag{B.39}$$

We invoke MATLAB, we start Simulink by clicking on the Simulink icon, on the **Simulink Library Browser** we click on the **Create a new model** (blank page icon on the left of the top bar), and we save this model as Example_1_2. On the **Simulink Library Browser** we select **Sources**, we drag the **Signal Generator** block on the Example_1_2 model window, we click and drag the **State–Space** block from the **Continuous** on **Simulink Library Browser**, and we click and drag the **Scope** block from the **Commonly Used Blocks** on the **Simulink Library Browser**. We also add the **Display** block found under **Sinks** on the **Simulink Library Browser**. We connect these four blocks and the complete block diagram is as shown in Figure B.17.

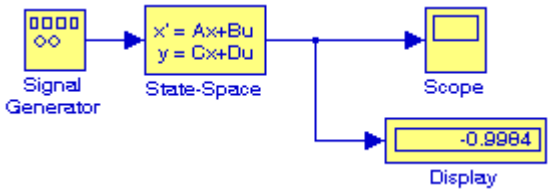


Figure B.17. Block diagram for Example B.2

We now double–click on the **Signal Generator** block and we enter the following in the **Function Block Parameters:**

Wave form: sine

Time (t): Use simulation time

Amplitude: 1

Frequency: 2

Units: Hertz

Next, we double–click on the **state–space** block and we enter the following parameter values in the **Function Block Parameters**:

A: [0 1 0 0; 0 0 1 0; 0 0 0 1; –a0 –a1 –a2 –a3]

B: [0 0 0 1]'

C: [1 0 0 0]

D: [0]

Initial conditions: x0

Absolute tolerance: auto

Now, we switch to the MATLAB Command prompt and we type the following:

```
>> a0=1; a1=0; a2=2; a3=0; x0=[0  0  0  0]';
```

We change the Simulation Stop time to 25, and we start the simulation by clicking on the  icon. To see the output waveform, we double click on the **Scope** block, then clicking on the Autoscale  icon, we obtain the waveform shown in Figure B.18.

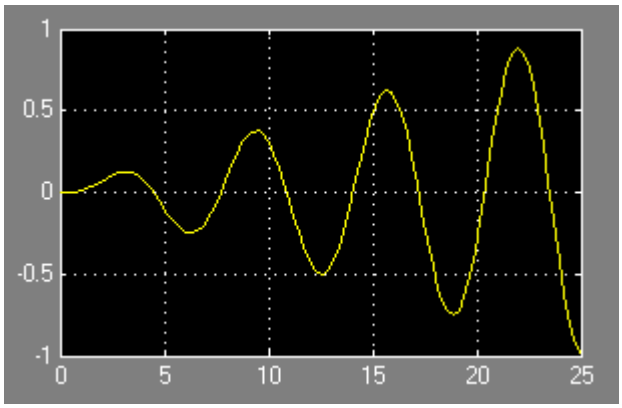


Figure B.18. Waveform for Example B.2

The **Display** block in Figure B.17 shows the value at the end of the simulation stop time.

Examples B.1 and B.2 have clearly illustrated that the State–Space is indeed a powerful block. We could have obtained the solution of Example B.2 using four Integrator blocks by this approach would have been more time consuming.

Example B.3

Using **Algebraic Constraint** blocks found in the **Math Operations** library, **Display** blocks found in the **Sinks** library, and **Gain** blocks found in the **Commonly Used Blocks** library, we will create a model that will produce the simultaneous solution of three equations with three unknowns.

The model will display the values for the unknowns z_1, z_2, and z_3 in the system of the equations

$$\begin{aligned} a_1z_1 + a_2z_2 + a_3z_3 + k_1 &= 0 \\ a_4z_1 + a_5z_2 + a_6z_3 + k_2 &= 0 \\ a_7z_1 + a_8z_2 + a_9z_3 + k_3 &= 0 \end{aligned} \tag{B.40}$$

The model is shown in Figure B.19.

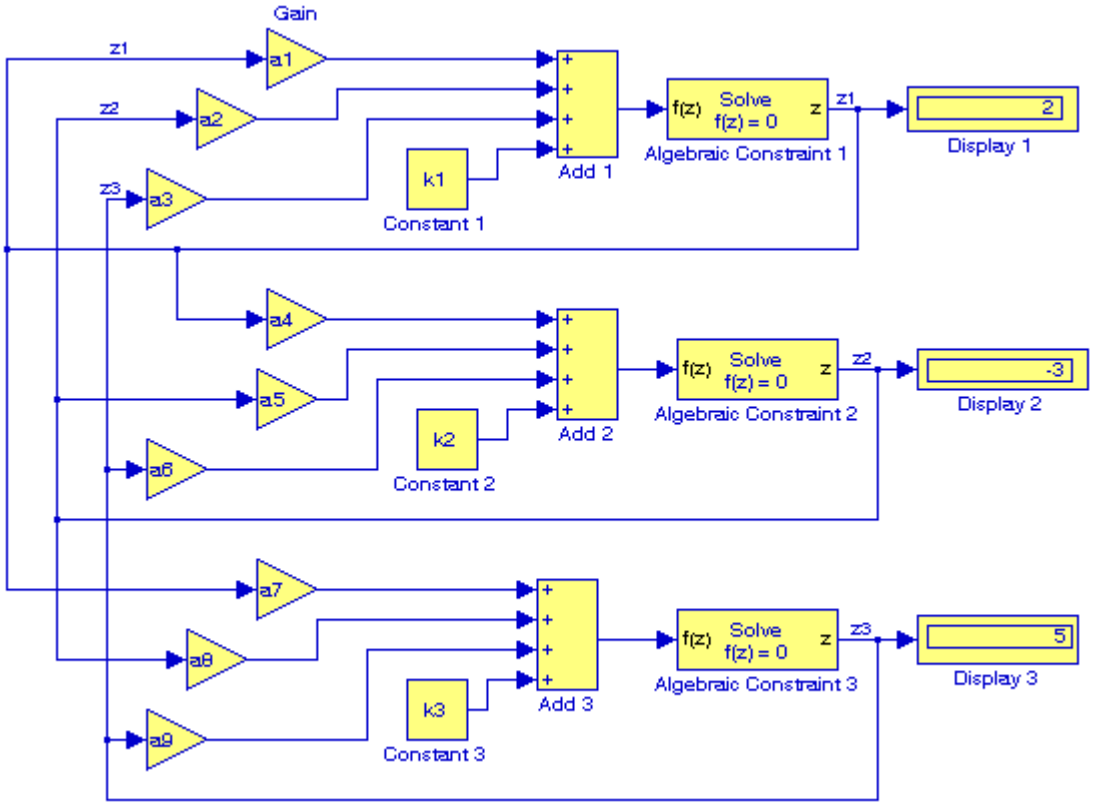


Figure B.19. Model for Example B.3

Next, we go to MATLAB's Command prompt and we enter the following values:

```
a1=2; a2=–3; a3=–1; a4=1; a5=5; a6=4; a7=–6; a8=1; a9=2;...
k1=–8; k2=–7; k3=5;
```

After clicking on the simulation icon, we observe the values of the unknowns as $z_1 = 2$, $z_2 = -3$, and $z_3 = 5$. These values are shown in the Display blocks of Figure B.19.

The **Algebraic Constraint** block constrains the input signal $f(z)$ to zero and outputs an algebraic state z. The block outputs the value necessary to produce a zero at the input. The output must affect the input through some feedback path. This enables us to specify algebraic equations for index 1 differential/algebraic systems (DAEs). By default, the Initial guess parameter is zero. We can improve the efficiency of the algebraic loop solver by providing an Initial guess for the algebraic state z that is close to the solution value.

An outstanding feature in Simulink is the representation of a large model consisting of many blocks and lines, to be shown as a single Subsystem block.* For instance, we can group all blocks and lines in the model of Figure B.19 except the display blocks, we choose **Create Subsystem** from the **Edit** menu, and this model will be shown as in Figure B.20† where in MATLAB's Command prompt we have entered:

```
a1=5; a2=–1; a3=4; a4=11; a5=6; a6=9; a7=–8; a8=4; a9=15;...
k1=14; k2=–6; k3=9;
```

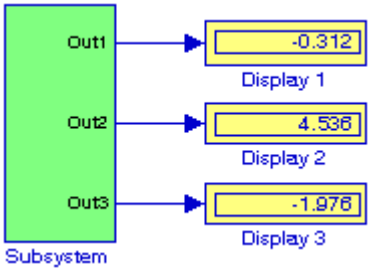


Figure B.20. The model of Figure B.19 represented as a subsystem

The Display blocks in Figure B.20 show the values of z_1, z_2, and z_3 for the values specified in MATLAB's Command prompt.

B.2 Simulink Demos

At this time, the reader with no prior knowledge of Simulink, should be ready to learn Simulink's additional capabilities. It is highly recommended that the reader becomes familiar with the block libraries found in the Simulink Library Browser. Then, the reader can follow the steps delineated in The MathWorks Simulink User's Manual to run the Demo Models beginning with the thermo model. This model can be seen by typing

```
thermo
```

in the MATLAB Command prompt.

* *The Subsystem block is described in detail in Chapter 2, Section 2.1, Page 2–2, Introduction to Simulink with Engineering Applications, ISBN 0–9744239–7–1.*

† *The contents of the Subsystem block are not lost. We can double–click on the Subsystem block to see its contents. The Subsystem block replaces the inputs and outputs of the model with Inport and Outport blocks. These blocks are described in Section 2.1, Chapter 2, Page 2–2, Introduction to Simulink with Engineering Applications, ISBN 0–9744239–7–1.*

Appendix C

A Review of Complex Numbers

This appendix is a review of the algebra of complex numbers. The basic operations are defined and illustrated with several examples. Applications using Euler's identities are presented, and the exponential and polar forms are discussed and illustrated with examples.

C.1 Definition of a Complex Number

In the language of mathematics, the square root of minus one is denoted as i, that is, $i = \sqrt{-1}$. In the electrical engineering field, we denote i as j to avoid confusion with current i. Essentially, j is an operator that produces a 90–degree counterclockwise rotation to any vector to which it is applied as a multiplying factor. Thus, if it is given that a vector A has the direction along the right side of the x–axis as shown in Figure C.1, multiplication of this vector by the operator j will result in a new vector jA whose magnitude remains the same, but it has been rotated counterclockwise by $90°$. Also, another multiplication of the new vector jA by j will produce another $90°$ counterclockwise direction. In this case, the vector A has rotated $180°$ and its new value now is $-A$. When this vector is rotated by another $90°$ for a total of $270°$, its value becomes $j(-A) = -jA$. A fourth $90°$ rotation returns the vector to its original position, and thus its value is again A. Therefore, we conclude that $j^2 = -1$, $j^3 = -j$, and $j^4 = 1$.

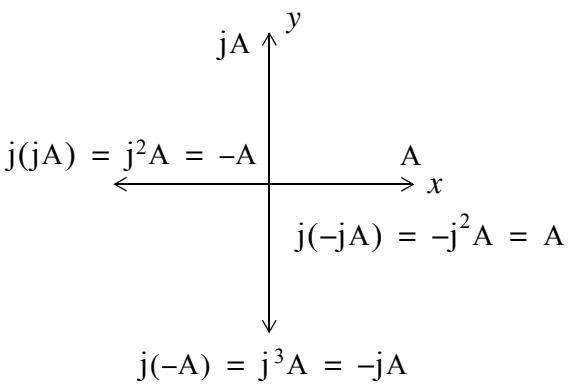


Figure C.1. The j operator

Note: In our subsequent discussion, we will designate the x–axis (abscissa) as the *real axis*, and the y–axis (ordinate) as the *imaginary axis* with the understanding that the "imaginary" axis is

just as "real" as the real axis. In other words, the imaginary axis is just as important as the real axis.*

An *imaginary number* is the product of a real number, say r, by the operator j. Thus, r is a real number and jr is an imaginary number.

A *complex number* is the sum (or difference) of a real number and an imaginary number. For example, the number $A = a + jb$ where a and b are both real numbers, is a complex number. Then, $a = \text{Re}\{A\}$ and $b = \text{Im}\{A\}$ where $\text{Re}\{A\}$ denotes real part of A, and $b = \text{Im}\{A\}$ the imaginary part of A.

By definition, two complex numbers A and B where $A = a + jb$ and $B = c + jd$, are equal if and only if their real parts are equal, and also their imaginary parts are equal. Thus, $A = B$ if and only if $a = c$ and $b = d$.

C.2 Addition and Subtraction of Complex Numbers

The sum of two complex numbers has a real component equal to the sum of the real components, and an imaginary component equal to the sum of the imaginary components. For subtraction, we change the signs of the components of the subtrahend and we perform addition. Thus, if

$$A = a + jb \text{ and } B = c + jd$$

then

$$A + B = (a + c) + j(b + d)$$

and

$$A - B = (a - c) + j(b - d)$$

Example C.1

It is given that $A = 3 + j4$, and $B = 4 - j2$. Find $A + B$ and $A - B$

Solution:

$$A + B = (3 + j4) + (4 - j2) = (3 + 4) + j(4 - 2) = 7 + j2$$

and

$$A - B = (3 + j4) - (4 - j2) = (3 - 4) + j(4 + 2) = -1 + j6$$

* *We may think the real axis as the cosine axis and the imaginary axis as the sine axis.*

C.3 Multiplication of Complex Numbers

Complex numbers are multiplied using the rules of elementary algebra, and making use of the fact that $j^2 = -1$. Thus, if

$$A = a + jb \text{ and } B = c + jd$$

then

$$A \cdot B = (a + jb) \cdot (c + jd) = ac + jad + jbc + j^2bd$$

and since $j^2 = -1$, it follows that

$$\begin{aligned} A \cdot B &= ac + jad + jbc - bd \\ &= (ac - bd) + j(ad + bc) \end{aligned} \tag{C.1}$$

Example C.2

It is given that $A = 3 + j4$ and $B = 4 - j2$. Find $A \cdot B$

Solution:

$$A \cdot B = (3 + j4) \cdot (4 - j2) = 12 - j6 + j16 - j^2 8 = 20 + j10$$

The *conjugate* of a complex number, denoted as A^*, is another complex number with the same real component, and with an imaginary component of opposite sign. Thus, if $A = a + jb$, then $A^* = a - jb$.

Example C.3

It is given that $A = 3 + j5$. Find A^*

Solution:

The conjugate of the complex number A has the same real component, but the imaginary component has opposite sign. Then, $A^* = 3 - j5$

If a complex number A is multiplied by its conjugate, the result is a real number. Thus, if $A = a + jb$, then

$$A \cdot A^* = (a + jb)(a - jb) = a^2 - jab + jab - j^2b^2 = a^2 + b^2$$

Example C.4

It is given that $A = 3 + j5$. Find $A \cdot A^*$

Solution:

$$A \cdot A^* = (3 + j5)(3 - j5) = 3^2 + 5^2 = 9 + 25 = 34$$

C.4 Division of Complex Numbers

When performing division of complex numbers, it is desirable to obtain the quotient separated into a real part and an imaginary part. This procedure is called *rationalization of the quotient*, and it is done by multiplying the denominator by its conjugate. Thus, if $A = a + jb$ and $B = c + jd$, then,

$$\frac{A}{B} = \frac{a + jb}{c + jd} = \frac{(a + jb)(c - jd)}{(c + jd)(c - jd)} = \frac{A}{B} \cdot \frac{B^*}{B^*} = \frac{(ac + bd) + j(bc - ad)}{c^2 + d^2}$$
$$= \frac{(ac + bd)}{c^2 + d^2} + j\frac{(bc - ad)}{c^2 + d^2} \qquad \text{(C.2)}$$

In (C.2), we multiplied both the numerator and denominator by the conjugate of the denominator to eliminate the *j* operator from the denominator of the quotient. Using this procedure, we see that the quotient is easily separated into a real and an imaginary part.

Example C.5

It is given that $A = 3 + j4$, and $B = 4 + j3$. Find A/B

Solution:

Using the procedure of (C.2), we obtain

$$\frac{A}{B} = \frac{3 + j4}{4 + j3} = \frac{(3 + j4)(4 - j3)}{(4 + j3)(4 - j3)} = \frac{12 - j9 + j16 + 12}{4^2 + 3^2} = \frac{24 + j7}{25} = \frac{24}{25} + j\frac{7}{25} = 0.96 + j0.28$$

C.5 Exponential and Polar Forms of Complex Numbers

The relations

$$\boxed{e^{j\theta} = \cos\theta + j\sin\theta} \qquad \text{(C.3)}$$

and

$$\boxed{e^{-j\theta} = \cos\theta - j\sin\theta} \tag{C.4}$$

are known as the *Euler's identities*.

Multiplying (C.3) by the *real* positive constant C we obtain:

$$Ce^{j\theta} = C\cos\theta + jC\sin\theta \tag{C.5}$$

This expression represents a complex number, say $a + jb$, and thus

$$Ce^{j\theta} = a + jb \tag{C.6}$$

where the left side of (C.6) is the *exponential form*, and the right side is the *rectangular form*.

Equating real and imaginary parts in (C.5) and (C.6), we obtain

$$a = C\cos\theta \quad \text{and} \quad b = C\sin\theta \tag{C.7}$$

Squaring and adding the expressions in (C.7), we obtain

$$a^2 + b^2 = (C\cos\theta)^2 + (C\sin\theta)^2 = C^2(\cos^2\theta + \sin^2\theta) = C^2$$

Then,

$$C^2 = a^2 + b^2$$

or

$$\boxed{C = \sqrt{a^2 + b^2}} \tag{C.8}$$

Also, from (C.7)

$$\frac{b}{a} = \frac{C\sin\theta}{C\cos\theta} = \tan\theta$$

or

$$\boxed{\theta = \tan^{-1}\left(\frac{b}{a}\right)} \tag{C.9}$$

To convert a complex number from rectangular to exponential form, we use the expression

$$\boxed{a + jb = \sqrt{a^2 + b^2}\, e^{j\left(\tan^{-1}\frac{b}{a}\right)}} \tag{C.10}$$

To convert a complex number from exponential to rectangular form, we use the expressions

$$\boxed{\begin{aligned} Ce^{j\theta} &= C\cos\theta + jC\sin\theta \\ Ce^{-j\theta} &= C\cos\theta - jC\sin\theta \end{aligned}} \tag{C.11}$$

The *polar form* is essentially the same as the exponential form but the notation is different, that is,

$$\boxed{Ce^{j\theta} = C\angle\theta} \tag{C.12}$$

where the left side of (C.12) is the exponential form, and the right side is the polar form.

We must remember that *the phase angle* θ *is always measured with respect to the positive real axis, and rotates in the counterclockwise direction.*

Example C.6

Convert the following complex numbers to exponential and polar forms:

a. $3 + j4$

b. $-1 + j2$

c. $-2 - j$

d. $4 - j3$

Solution:

a. The real and imaginary components of this complex number are shown in Figure C.2.

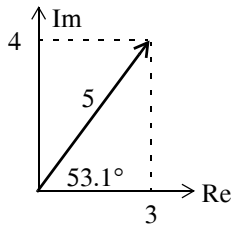


Figure C.2. The components of $3 + j4$

Then,

$$3 + j4 = \sqrt{3^2 + 4^2}\, e^{j\left(\tan^{-1}\frac{4}{3}\right)} = 5e^{j53.1°} = 5\angle 53.1°$$

Check with MATLAB:

```
x=3+j*4; magx=abs(x); thetax=angle(x)*180/pi;  disp(magx); disp(thetax)

    5
   53.1301
```

Check with the Simulink **Complex to Magnitude–Angle** block* shown in the Simulink model of Figure C.3.

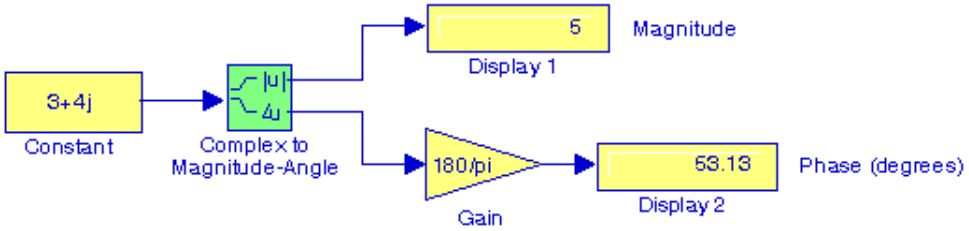


Figure C.3. Simulink model for Example C.6a

b. The real and imaginary components of this complex number are shown in Figure C.4.

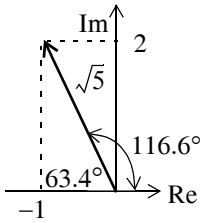


Figure C.4. The components of $-1+j2$

Then,

$$-1+j2 = \sqrt{1^2+2^2}\,e^{j\left(\tan^{-1}\frac{2}{-1}\right)} = \sqrt{5}e^{j116.6^\circ} = \sqrt{5}\angle 116.6^\circ$$

Check with MATLAB:

```
y=−1+j*2; magy=abs(y); thetay=angle(y)*180/pi;  disp(magy); disp(thetay)

    2.2361
  116.5651
```

c. The real and imaginary components of this complex number are shown in Figure C.5.

* *For a detailed description and examples with this and other related transformation blocks, please refer to Introduction to Simulink with Engineering Applications, ISBN 0–9744239–7–1, Section 8.3, Chapter 8, Page 8–24, and Section 19.8, Chapter 19, Page 19–27.*

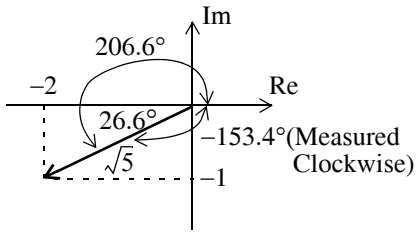


Figure C.5. The components of $-2-j$

Then,

$$-2-j1 = \sqrt{2^2+1^2}\,e^{j\left(\tan^{-1}\frac{-1}{-2}\right)} = \sqrt{5}e^{j206.6^\circ} = \sqrt{5}\angle 206.6^\circ = \sqrt{5}e^{j(-153.4)^\circ} = \sqrt{5}\angle -153.4^\circ$$

Check with MATLAB:

```
v=–2–j*1; magv=abs(v); thetav=angle(v)*180/pi;  disp(magv); disp(thetav)

    2.2361
 -153.4349
```

d. The real and imaginary components of this complex number are shown in Figure C.5.

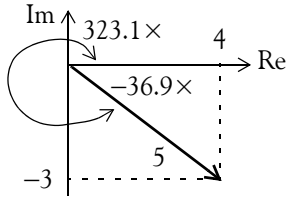


Figure C.6. The components of $4-j3$

Then,

$$4-j3 = \sqrt{4^2+3^2}\,e^{j\left(\tan^{-1}\frac{-3}{4}\right)} = 5e^{j323.1^\circ} = 5\angle 323.1^\circ = 5e^{-j36.9^\circ} = 5\angle -36.9^\circ$$

Check with MATLAB:

```
w=4–j*3; magw=abs(w); thetaw=angle(w)*180/pi;  disp(magw); disp(thetaw)

     5
 -36.8699
```

Example C.7

Express the complex number $-2\angle 30°$ in exponential and in rectangular forms.

Solution:

We recall that $-1 = j^2$. Since each j rotates a vector by $90°$ counterclockwise, then $-2\angle 30°$ is the same as $2\angle 30°$ rotated counterclockwise by $180°$. Therefore,

$$-2\angle 30° = 2\angle(30° + 180°) = 2\angle 210° = 2\angle -150°$$

The components of this complex number are shown in Figure C.6.

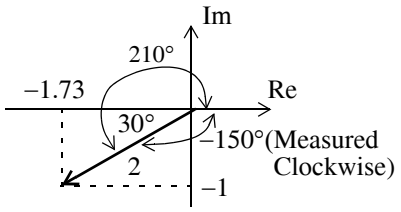


Figure C.7. The components of $2\angle -150°$

Then,

$$2\angle -150° = 2e^{-j150°} = 2(\cos 150° - j\sin 150°) = 2(-0.866 - j0.5) = -1.73 - j$$

Note: The rectangular form is most useful when we add or subtract complex numbers; however, the exponential and polar forms are most convenient when we multiply or divide complex numbers.

To multiply two complex numbers in exponential (or polar) form, we multiply the magnitudes and we add the phase angles, that is, if

$$A = M\angle\theta \quad \text{and} \quad B = N\angle\phi$$

then,

$$\boxed{AB = MN\angle(\theta + \phi) = Me^{j\theta}Ne^{j\phi} = MNe^{j(\theta + \phi)}} \qquad (C.13)$$

Example C.8

Multiply $A = 10\angle 53.1°$ by $B = 5\angle -36.9°$

Solution:

Multiplication in polar form yields

$$AB = (10 \times 5)\angle[53.1° + (-36.9°)] = 50\angle 16.2°$$

and multiplication in exponential form yields

$$AB = (10e^{j53.1°})(5e^{-j36.9°}) = 50e^{j(53.1° - 36.9°)} = 50e^{j16.2°}$$

To divide one complex number by another when both are expressed in exponential or polar form, we divide the magnitude of the dividend by the magnitude of the divisor, and we subtract the phase angle of the divisor from the phase angle of the dividend, that is, if

$$A = M\angle\theta \quad \text{and} \quad B = N\angle\phi$$

then,

$$\frac{A}{B} = \frac{M}{N}\angle(\theta - \phi) = \frac{Me^{j\theta}}{Ne^{j\phi}} = \frac{M}{N}e^{j(\theta - \phi)} \tag{C.14}$$

Example C.9

Divide $A = 10\angle 53.1°$ by $B = 5\angle -36.9°$

Solution:

Division in polar form yields

$$\frac{A}{B} = \frac{10\angle 53.1°}{5\angle -36.9°} = 2\angle[53.1° - (-36.9°)] = 2\angle 90°$$

Division in exponential form yields

$$\frac{A}{B} = \frac{10e^{j53.1°}}{5e^{-j36.9°}} = 2e^{j53.1°}e^{j36.9°} = 2e^{j90°}$$

Appendix D

Matrices and Determinants

This appendix is an introduction to matrices and matrix operations. Determinants, Cramer's rule, and Gauss's elimination method are reviewed. Some definitions and examples are not applicable to the material presented in this text, but are included for subject continuity, and academic interest. They are discussed in detail in matrix theory textbooks. These are denoted with a dagger (†) and may be skipped.

D.1 Matrix Definition

A *matrix* is a rectangular array of numbers such as those shown below.

$$\begin{bmatrix} 2 & 3 & 7 \\ 1 & -1 & 5 \end{bmatrix} \qquad \text{or} \qquad \begin{bmatrix} 1 & 3 & 1 \\ -2 & 1 & -5 \\ 4 & -7 & 6 \end{bmatrix}$$

In general form, a matrix A is denoted as

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} \tag{D.1}$$

The numbers a_{ij} are the *elements* of the matrix where the index i indicates the row, and j indicates the column in which each element is positioned. For instance, a_{43} indicates the element positioned in the fourth row and third column.

A matrix of m rows and n columns is said to be of $m \times n$ *order matrix*.

If $m = n$, the matrix is said to be a *square matrix of order* m (or n). Thus, if a matrix has five rows and five columns, it is said to be a square matrix of order 5.

In a square matrix, the elements a_{11}, a_{22}, a_{33}, …, a_{nn} are called the *main diagonal elements*. Alternately, we say that the matrix elements a_{11}, a_{22}, a_{33}, …, a_{nn}, are located on the *main diagonal*.

† The sum of the diagonal elements of a square matrix A is called the *trace*[*] *of* A.

† A matrix in which every element is zero, is called a *zero matrix*.

D.2 Matrix Operations

Two matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ are equal, that is, $A = B$, if and only if

$$a_{ij} = b_{ij} \qquad i = 1, 2, 3, \ldots, m \qquad j = 1, 2, 3, \ldots, n \tag{D.2}$$

Two matrices are said to be *conformable for addition* (*subtraction*), if they are of the same order $m \times n$.

If $A = [a_{ij}]$ and $B = [b_{ij}]$ are conformable for addition (subtraction), their sum (difference) will be another matrix C with the same order as A and B, where each element of C is the sum (difference) of the corresponding elements of A and B, that is,

$$C = A \pm B = [a_{ij} \pm b_{ij}] \tag{D.3}$$

Example D.1

Compute $A + B$ and $A - B$ given that

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & 3 & 0 \\ -1 & 2 & 5 \end{bmatrix}$$

Solution:

$$A + B = \begin{bmatrix} 1+2 & 2+3 & 3+0 \\ 0-1 & 1+2 & 4+5 \end{bmatrix} = \begin{bmatrix} 3 & 5 & 3 \\ -1 & 3 & 9 \end{bmatrix}$$

and

* *Henceforth, all paragraphs and topics preceded by a dagger (†) may be skipped. These are discussed in matrix theory textbooks.*

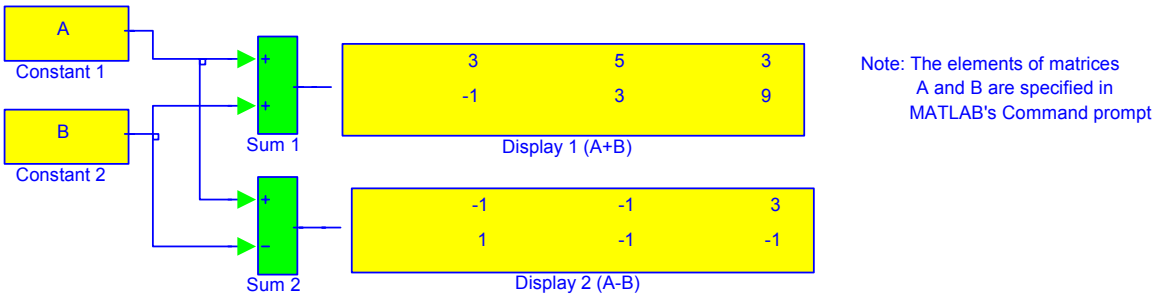
$$A - B = \begin{bmatrix} 1-2 & 2-3 & 3-0 \\ 0+1 & 1-2 & 4-5 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 3 \\ 1 & -1 & -1 \end{bmatrix}$$

Check with MATLAB:

```
A=[1  2  3;   0  1  4];  B=[2  3  0; –1  2  5];    % Define matrices A and B
A+B, A–B                                           % Add A and B, then Subtract B from A
```

```
ans =
     3     5     3
    -1     3     9

ans =
    -1    -1     3
     1    -1    -1
```

Check with Simulink:



If k is any scalar (a positive or negative number), and not [k] which is a 1×1 matrix, then multiplication of a matrix A by the scalar k is the multiplication of every element of A by k.

Example D.2

Multiply the matrix

$$A = \begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix}$$

by

a. $k_1 = 5$

b. $k_2 = -3 + j2$

Solution:

a.

$$k_1 \cdot A = 5 \times \begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 5 \times 1 & 5 \times (-2) \\ 5 \times 2 & 5 \times 3 \end{bmatrix} = \begin{bmatrix} 5 & -10 \\ 10 & 15 \end{bmatrix}$$

b.

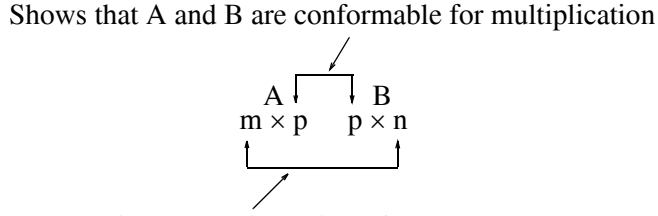
$$k_2 \cdot A = (-3 + j2) \times \begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} (-3 + j2) \times 1 & (-3 + j2) \times (-2) \\ (-3 + j2) \times 2 & (-3 + j2) \times 3 \end{bmatrix} = \begin{bmatrix} -3 + j2 & 6 - j4 \\ -6 + j4 & -9 + j6 \end{bmatrix}$$

Check with MATLAB:

```
k1=5; k2=(–3 + 2*j);          %  Define scalars k1 and k2
A=[1 –2; 2  3];               % Define matrix A
k1*A, k2*A                    % Multiply matrix A by scalars k1 and k2
```

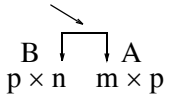
```
ans =
     5   -10
    10    15

ans =
  -3.0000+ 2.0000i   6.0000- 4.0000i
  -6.0000+ 4.0000i  -9.0000+ 6.0000i
```

Two matrices A and B are said to be *conformable for multiplication* $A \cdot B$ in that order, only when the number of columns of matrix A is equal to the number of rows of matrix B. That is, the product $A \cdot B$ (but not $B \cdot A$) is conformable for multiplication only if A is an $m \times p$ matrix and matrix B is an $p \times n$ matrix. The product $A \cdot B$ will then be an $m \times n$ matrix. A convenient way to determine if two matrices are conformable for multiplication is to write the dimensions of the two matrices side–by–side as shown below.



For the product $B \cdot A$ we have:

Here, B and A are not conformable for multiplication



For matrix multiplication, the operation is row by column. Thus, to obtain the product $A \cdot B$, we multiply each element of a row of A by the corresponding element of a column of B; then, we add these products.

Example D.3

Matrices C and D are defined as

$$C = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} \text{ and } D = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$$

Compute the products $C \cdot D$ and $D \cdot C$

Solution:

The dimensions of matrices C and D are respectively 1×3 3×1; therefore the product $C \cdot D$ is feasible, and will result in a 1×1, that is,

$$C \cdot D = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} (2) \cdot (1) + (3) \cdot (-1) + (4) \cdot (2) \end{bmatrix} = \begin{bmatrix} 7 \end{bmatrix}$$

The dimensions for D and C are respectively 3×1 1×3 and therefore, the product $D \cdot C$ is also feasible. Multiplication of these will produce a 3×3 matrix as follows:

$$D \cdot C = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} = \begin{bmatrix} (1) \cdot (2) & (1) \cdot (3) & (1) \cdot (4) \\ (-1) \cdot (2) & (-1) \cdot (3) & (-1) \cdot (4) \\ (2) \cdot (2) & (2) \cdot (3) & (2) \cdot (4) \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \\ -2 & -3 & -4 \\ 4 & 6 & 8 \end{bmatrix}$$

Check with MATLAB:

```
C=[2  3  4];  D=[1  −1  2]';     % Define matrices C and D. Observe that D is a column vector
C*D, D*C                         % Multiply C by D, then multiply D by C

  ans =
       7
```

```
ans =
     2     3     4
    -2    -3    -4
     4     6     8
```

Division of one matrix by another, is not defined. However, an analogous operation exists, and it will become apparent later in this chapter when we discuss the inverse of a matrix.

D.3 Special Forms of Matrices

† A square matrix is said to be *upper triangular* when all the elements below the diagonal are zero. The matrix A of (D.4) is an upper triangular matrix. In an upper triangular matrix, not all elements above the diagonal need to be non–zero.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a_{22} & a_{23} & \dots & a_{2n} \\ 0 & 0 & \ddots & \dots & \dots \\ \dots & \dots & 0 & \ddots & \dots \\ 0 & 0 & 0 & \dots & a_{mn} \end{bmatrix} \tag{D.4}$$

† A square matrix is said to be *lower triangular*, when all the elements above the diagonal are zero. The matrix B of (D.5) is a lower triangular matrix. In a lower triangular matrix, not all elements below the diagonal need to be non–zero.

$$B = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & \dots & 0 \\ \dots & \dots & \ddots & 0 & 0 \\ \dots & \dots & \dots & \ddots & 0 \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} \tag{D.5}$$

† A square matrix is said to be *diagonal*, if all elements are zero, except those in the diagonal. The matrix C of (D.6) is a diagonal matrix.

$$C = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ 0 & a_{22} & 0 & \dots & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \dots & a_{mn} \end{bmatrix} \tag{D.6}$$

† A diagonal matrix is called a *scalar matrix*, if $a_{11} = a_{22} = a_{33} = \dots = a_{nn} = k$ where k is a scalar. The matrix D of (D.7) is a scalar matrix with $k = 4$.

$$D = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \tag{D.7}$$

A scalar matrix with $k = 1$, is called an *identity matrix* I. Shown below are 2×2, 3×3, and 4×4 identity matrices.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{D.8}$$

The MATLAB **eye(n)** function displays an $n \times n$ identity matrix. For example,

```
eye(4)     % Display a 4 by 4 identity matrix
```

```
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

Likewise, the **eye(size(A))** function, produces an identity matrix whose size is the same as matrix A. For example, let matrix A be defined as

```
A=[1  3  1; –2  1 –5; 4 –7  6]      % Define matrix A
```

```
A =
     1     3     1
    -2     1    -5
     4    -7     6
```

Then,

```
eye(size(A))
```

displays

```
ans =
     1     0     0
     0     1     0
     0     0     1
```

† The *transpose of a matrix* A, denoted as A^T, is the matrix that is obtained when the rows and columns of matrix A are interchanged. For example, if

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \text{then} \quad A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \tag{D.9}$$

In MATLAB, we use the apostrophe (') symbol to denote and obtain the transpose of a matrix. Thus, for the above example,

```
A=[1  2  3;  4  5  6]      % Define matrix A
A =
     1     2     3
     4     5     6
A'                         % Display the transpose of A
ans =
     1     4
     2     5
     3     6
```

† A *symmetric matrix* A is a matrix such that $A^T = A$, that is, the transpose of a matrix A is the same as A. An example of a symmetric matrix is shown below.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -5 \\ 3 & -5 & 6 \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -5 \\ 3 & -5 & 6 \end{bmatrix} = A \tag{D.10}$$

† If a matrix A has complex numbers as elements, the matrix obtained from A by replacing each element by its conjugate, is called the *conjugate of* A, and it is denoted as A*, for example,

$$A = \begin{bmatrix} 1+j2 & j \\ 3 & 2-j3 \end{bmatrix} \qquad A^* = \begin{bmatrix} 1-j2 & -j \\ 3 & 2+j3 \end{bmatrix}$$

MATLAB has two built–in functions which compute the complex conjugate of a number. The first, **conj(x)**, computes the complex conjugate of any complex number, and the second, **conj(A)**, computes the conjugate of a matrix A. Using MATLAB with the matrix A defined as above, we obtain

```
A = [1+2j   j;  3   2–3j]  % Define and display matrix A

A =
  1.0000 + 2.0000i        0 + 1.0000i
  3.0000                2.0000 - 3.0000i

conj_A=conj(A)          % Compute and display the conjugate of A

conj_A =
  1.0000 - 2.0000i        0 - 1.0000i
  3.0000                2.0000 + 3.0000i
```

† A square matrix A such that $A^T = -A$ is called *skew-symmetric*. For example,

$$A = \begin{bmatrix} 0 & 2 & -3 \\ -2 & 0 & -4 \\ 3 & 4 & 0 \end{bmatrix} \quad A^T = \begin{bmatrix} 0 & -2 & 3 \\ 2 & 0 & 4 \\ -3 & -4 & 0 \end{bmatrix} = -A$$

Therefore, matrix A above is skew symmetric.

† A square matrix A such that $A^{T*} = A$ is called *Hermitian*. For example,

$$A = \begin{bmatrix} 1 & 1-j & 2 \\ 1+j & 3 & j \\ 2 & -j & 0 \end{bmatrix} \quad A^T = \begin{bmatrix} 1 & 1+j & 2 \\ 1-j & 3 & -j \\ 2 & j & 0 \end{bmatrix} \quad A^{T*} = \begin{bmatrix} 1 & 1+j & 2 \\ 1-j & 3 & -j \\ 2 & j & 0 \end{bmatrix} = A$$

Therefore, matrix A above is Hermitian.

† A square matrix A such that $A^{T*} = -A$ is called *skew–Hermitian*. For example,

$$A = \begin{bmatrix} j & 1-j & 2 \\ -1-j & 3j & j \\ -2 & j & 0 \end{bmatrix} \quad A^T = \begin{bmatrix} j & -1-j & -2 \\ 1-j & 3j & j \\ 2 & j & 0 \end{bmatrix} \quad A^{T*} = \begin{bmatrix} -j & -1+j & -2 \\ 1+j & -3j & -j \\ 2 & -j & 0 \end{bmatrix} = -A$$

Therefore, matrix A above is skew–Hermitian.

D.4 Determinants

Let matrix A be defined as the square matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \tag{D.11}$$

then, the *determinant of* A, denoted as detA, is defined as

$$\begin{aligned} \det A = {} & a_{11}a_{22}a_{33}\ldots a_{nn} + a_{12}a_{23}a_{34}\ldots a_{n1} + a_{13}a_{24}a_{35}\ldots a_{n2} + \ldots \\ & -a_{n1}\ldots a_{22}a_{13}\ldots -a_{n2}\ldots a_{23}a_{14} - a_{n3}\ldots a_{24}a_{15} - \ldots \end{aligned} \tag{D.12}$$

The determinant of a square matrix of order *n* is referred to as *determinant of order n.*

Let A be a determinant of order 2, that is,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{D.13}$$

Then,

$$\det A = a_{11}a_{22} - a_{21}a_{12} \tag{D.14}$$

Example D.4

Matrices A and B are defined as

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & -1 \\ 2 & 0 \end{bmatrix}$$

Compute detA and detB.

Solution:

$$\begin{aligned} \det A &= 1 \cdot 4 - 3 \cdot 2 = 4 - 6 = -2 \\ \det B &= 2 \cdot 0 - 2 \cdot (-1) = 0 - (-2) = 2 \end{aligned}$$

Check with MATLAB:

```
A=[1  2; 3  4]; B=[2  −1; 2  0];        % Define matrices A and B
det(A), det(B)                          % Compute the determinants of A and B
```

```
ans =
    -2

ans =
     2
```

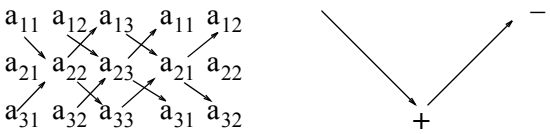
Let A be a matrix of order 3, that is,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{D.15}$$

then, detA is found from

$$\begin{aligned} \det A = {} & a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{11}a_{22}a_{33} \\ & -a_{11}a_{22}a_{33} - a_{11}a_{22}a_{33} - a_{11}a_{22}a_{33} \end{aligned} \tag{D.16}$$

A convenient method to evaluate the determinant of order 3, is to write the first two columns to the right of the 3×3 matrix, and add the products formed by the diagonals from upper left to lower right; then subtract the products formed by the diagonals from lower left to upper right as shown on the diagram of the next page. When this is done properly, we obtain (D.16) above.



This method works only with second and third order determinants. To evaluate higher order determinants, we must first compute the *cofactors*; these will be defined shortly.

Example D.5

Compute detA and detB if matrices A and B are defined as

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 2 & -3 & -4 \\ 1 & 0 & -2 \\ 0 & -5 & -6 \end{bmatrix}$$

Solution:

$$\text{detA} = \begin{matrix} 2 & 3 & 5 & 2 & 3 \\ 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 2 & 1 \end{matrix}$$

or

$$\text{detA} = (2\times 0\times 0)+(3\times 1\times 1)+(5\times 1\times 1) - (2\times 0\times 5)-(1\times 1\times 2)-(0\times 1\times 3) = 11-2=9$$

Likewise,

$$\text{detB} = \begin{matrix} 2 & -3 & -4 & 2 & -3 \\ 1 & 0 & -2 & 1 & -2 \\ 0 & -5 & -6 & 2 & -6 \end{matrix}$$

or

$$\text{detB} = [2\times 0\times(-6)]+[(-3)\times(-2)\times 0]+[(-4)\times 1\times(-5)] - [0\times 0\times(-4)]-[(-5)\times(-2)\times 2]-[(-6)\times 1\times(-3)] = 20-38=-18$$

Check with MATLAB:

```
A=[2  3  5; 1  0  1; 2  1  0]; det(A)        % Define matrix A and compute detA
```

```
  ans =
       9
```

```
B=[2  -3  -4; 1  0  -2; 0  -5  -6];det(B) % Define matrix B and compute detB
```

```
  ans =
     -18
```

D.5 Minors and Cofactors

Let matrix A be defined as the square matrix of order n as shown below.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \tag{D.17}$$

If we remove the elements of its ith row, and jth column, the remaining $n-1$ square matrix is called the *minor of* A, and it is denoted as $\left[M_{ij}\right]$.

The signed minor $(-1)^{i+j}\left[M_{ij}\right]$ is called the *cofactor* of a_{ij} and it is denoted as α_{ij}.

Example D.6

Matrix A is defined as

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{D.18}$$

Compute the minors $\left[M_{11}\right]$, $\left[M_{12}\right]$, $\left[M_{13}\right]$ and the cofactors α_{11}, α_{12} and α_{13}.

Solution:

$$\left[M_{11}\right] = \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix} \qquad \left[M_{12}\right] = \begin{bmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix} \qquad \left[M_{11}\right] = \begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

and

$$\alpha_{11} = (-1)^{1+1}\left[M_{11}\right] = \left[M_{11}\right] \qquad \alpha_{12} = (-1)^{1+2}\left[M_{12}\right] = -\left[M_{12}\right] \qquad \alpha_{13} = \left[M_{13}\right] = (-1)^{1+3}\left[M_{13}\right]$$

The remaining minors

$$\left[M_{21}\right], \quad \left[M_{22}\right], \quad \left[M_{23}\right], \quad \left[M_{31}\right], \quad \left[M_{32}\right], \quad \left[M_{33}\right]$$

and cofactors

$$\alpha_{21}, \alpha_{22}, \alpha_{23}, \alpha_{31}, \alpha_{32}, \text{and } \alpha_{33}$$

are defined similarly.

Example D.7

Compute the cofactors of matrix A defined as

$$\mathrm{A} = \begin{bmatrix} 1 & 2 & -3 \\ 2 & -4 & 2 \\ -1 & 2 & -6 \end{bmatrix} \tag{D.19}$$

Solution:

$$\alpha_{11} = (-1)^{1+1}\begin{bmatrix} -4 & 2 \\ 2 & -6 \end{bmatrix} = 20 \qquad \alpha_{12} = (-1)^{1+2}\begin{bmatrix} 2 & 2 \\ -1 & -6 \end{bmatrix} = 10 \tag{D.20}$$

$$\alpha_{13} = (-1)^{1+3}\begin{bmatrix} 2 & -4 \\ -1 & 2 \end{bmatrix} = 0 \qquad \alpha_{21} = (-1)^{2+1}\begin{bmatrix} 2 & -3 \\ 2 & -6 \end{bmatrix} = 6 \tag{D.21}$$

$$\alpha_{22} = (-1)^{2+2}\begin{bmatrix} 1 & -3 \\ -1 & -6 \end{bmatrix} = -9 \qquad \alpha_{23} = (-1)^{2+3}\begin{bmatrix} 1 & 2 \\ -1 & 2 \end{bmatrix} = -4 \tag{D.22}$$

$$\alpha_{31} = (-1)^{3+1}\begin{bmatrix} 2 & -3 \\ -4 & 2 \end{bmatrix} = -8, \qquad \alpha_{32} = (-1)^{3+2}\begin{bmatrix} 1 & -3 \\ 2 & 2 \end{bmatrix} = -8 \tag{D.23}$$

$$\alpha_{33} = (-1)^{3+3}\begin{bmatrix} 1 & 2 \\ 2 & -4 \end{bmatrix} = -8 \tag{D.24}$$

It is useful to remember that the signs of the cofactors follow the pattern below

$$\begin{array}{ccccc} + & - & + & - & + \\ - & + & - & + & - \\ + & - & + & - & + \\ - & + & - & + & - \\ + & - & + & - & + \end{array}$$

that is, the cofactors on the diagonals have the same sign as their minors.

Let A be a square matrix of any size; the value of the determinant of A is the sum of the products obtained by multiplying each element of *any* row or *any* column by its cofactor.

Example D.8

Matrix A is defined as

$$A = \begin{bmatrix} 1 & 2 & -3 \\ 2 & -4 & 2 \\ -1 & 2 & -6 \end{bmatrix} \tag{D.25}$$

Compute the determinant of A using the elements of the first row.

Solution:

$$\det A = 1\begin{bmatrix} -4 & 2 \\ 2 & -6 \end{bmatrix} - 2\begin{bmatrix} 2 & 2 \\ -1 & -6 \end{bmatrix} - 3\begin{bmatrix} 2 & -4 \\ -1 & 2 \end{bmatrix} = 1 \times 20 - 2 \times (-10) - 3 \times 0 = 40$$

Check with MATLAB:

```
A=[1  2  –3; 2  –4  2; –1  2  –6]; det(A)        % Define matrix A and compute detA

ans =
    40
```

We must use the above procedure to find the determinant of a matrix A of order 4 or higher. Thus, a fourth-order determinant can first be expressed as the sum of the products of the elements of its first row by its cofactor as shown below.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = a_{11}\begin{bmatrix} a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} \end{bmatrix} - a_{21}\begin{bmatrix} a_{12} & a_{13} & a_{14} \\ a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} \end{bmatrix} \tag{D.26}$$

$$+a_{31}\begin{bmatrix} a_{12} & a_{13} & a_{14} \\ a_{22} & a_{23} & a_{24} \\ a_{42} & a_{43} & a_{44} \end{bmatrix} - a_{41}\begin{bmatrix} a_{12} & a_{13} & a_{14} \\ a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \end{bmatrix}$$

Determinants of order five or higher can be evaluated similarly.

Example D.9

Compute the value of the determinant of the matrix A defined as

$$A = \begin{bmatrix} 2 & -1 & 0 & -3 \\ -1 & 1 & 0 & -1 \\ 4 & 0 & 3 & -2 \\ -3 & 0 & 0 & 1 \end{bmatrix} \tag{D.27}$$

Solution:

Using the above procedure, we will multiply each element of the first column by its cofactor. Then,

$$\underbrace{A=2\begin{bmatrix} 1 & 0 & -1 \\ 0 & 3 & -2 \\ 0 & 0 & 1 \end{bmatrix}}_{[a]} \quad \underbrace{-(-1)\begin{bmatrix} -1 & 0 & -3 \\ 0 & 3 & -2 \\ 0 & 0 & 1 \end{bmatrix}}_{[b]} \quad \underbrace{+4\begin{bmatrix} -1 & 0 & -3 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}}_{[c]} \quad \underbrace{-(-3)\begin{bmatrix} -1 & 0 & -3 \\ 1 & 0 & -1 \\ 0 & 3 & -2 \end{bmatrix}}_{[d]}$$

Next, using the procedure of Example D.5 or Example D.8, we find

$$[a] = 6\ ,\ [b] = -3\ ,\ [c] = 0\ ,\ [d] = -36$$

and thus

$$\det A = [a] + [b] + [c] + [d] = 6 - 3 + 0 - 36 = -33$$

We can verify our answer with MATLAB as follows:

```
A=[ 2 –1  0 –3; –1  1  0 –1; 4  0  3 –2;  –3  0  0  1]; delta = det(A)
```

```
delta =
   -33
```

Some useful properties of determinants are given below.

Property 1: *If all elements of one row or one column are zero, the determinant is zero.* An example of this is the determinant of the cofactor [c] above.

Property 2: *If all the elements of one row or column are m times the corresponding elements of another row or column, the determinant is zero.* For example, if

$$A = \begin{bmatrix} 2 & 4 & 1 \\ 3 & 6 & 1 \\ 1 & 2 & 1 \end{bmatrix} \tag{D.28}$$

then,

$$\det A = \begin{vmatrix} 2 & 4 & 1 \\ 3 & 6 & 1 \\ 1 & 2 & 1 \end{vmatrix} \begin{matrix} 2 & 4 \\ 3 & 6 \\ 1 & 2 \end{matrix} = 12 + 4 + 6 - 6 - 4 - 12 = 0 \tag{D.29}$$

Here, detA is zero because the second column in A is 2 times the first column.

Check with MATLAB:

```
A=[2 4 1; 3 6 1; 1 2 1]; det(A)

ans =
     0
```

Property 3: *If two rows or two columns of a matrix are identical, the determinant is zero.* This follows from Property 2 with $m = 1$.

D.6 Cramer's Rule

Let us consider the systems of the three equations below

$$\begin{aligned} a_{11}x + a_{12}y + a_{13}z &= A \\ a_{21}x + a_{22}y + a_{23}z &= B \\ a_{31}x + a_{32}y + a_{33}z &= C \end{aligned} \tag{D.30}$$

and let

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad D_1 = \begin{vmatrix} A & a_{11} & a_{13} \\ B & a_{21} & a_{23} \\ C & a_{31} & a_{33} \end{vmatrix} \quad D_2 = \begin{vmatrix} a_{11} & A & a_{13} \\ a_{21} & B & a_{23} \\ a_{31} & C & a_{33} \end{vmatrix} \quad D_3 = \begin{vmatrix} a_{11} & a_{12} & A \\ a_{21} & a_{22} & B \\ a_{31} & a_{32} & C \end{vmatrix}$$

Cramer's rule states that the unknowns *x, y,* and *z* can be found from the relations

$$x = \frac{D_1}{\Delta} \qquad y = \frac{D_2}{\Delta} \qquad z = \frac{D_3}{\Delta} \tag{D.31}$$

provided that the determinant Δ (delta) is not zero.

We observe that the numerators of (D.31) are determinants that are formed from Δ by the substitution of the known values A, B, and C, for the coefficients of the desired unknown.

Cramer's rule applies to systems of two or more equations.

If (D.30) is a homogeneous set of equations, that is, if $A = B = C = 0$, then, D_1, D_2, and D_3 are all zero as we found in Property 1 above. Then, $x = y = z = 0$ also.

Example D.10

Use Cramer's rule to find v_1, v_2, and v_3 if

$$\begin{aligned} 2v_1 - 5 - v_2 + 3v_3 &= 0 \\ -2v_3 - 3v_2 - 4v_1 &= 8 \\ v_2 + 3v_1 - 4 - v_3 &= 0 \end{aligned} \tag{D.32}$$

and verify your answers with MATLAB.

Solution:

Rearranging the unknowns v, and transferring known values to the right side, we obtain

$$\begin{aligned} 2v_1 - v_2 + 3v_3 &= 5 \\ -4v_1 - 3v_2 - 2v_3 &= 8 \\ 3v_1 + v_2 - v_3 &= 4 \end{aligned} \tag{D.33}$$

By Cramer's rule,

$$\Delta = \begin{vmatrix} 2 & -1 & 3 \\ -4 & -3 & -2 \\ 3 & 1 & -1 \end{vmatrix} \begin{matrix} 2 & -1 \\ -4 & -3 \\ 3 & 1 \end{matrix} = 6 + 6 - 12 + 27 + 4 + 4 = 35$$

$$D_1 = \begin{vmatrix} 5 & -1 & 3 \\ 8 & -3 & -2 \\ 4 & 1 & -1 \end{vmatrix} \begin{matrix} 5 & -1 \\ 8 & -3 \\ 4 & 1 \end{matrix} = 15 + 8 + 24 + 36 + 10 - 8 = 85$$

$$D_2 = \begin{vmatrix} 2 & 5 & 3 \\ -4 & 8 & -2 \\ 3 & 4 & -1 \end{vmatrix} \begin{matrix} 2 & 5 \\ -4 & 8 \\ 3 & 4 \end{matrix} = -16 - 30 - 48 - 72 + 16 - 20 = -170$$

$$D_3 = \begin{vmatrix} 2 & -1 & 5 \\ -4 & -3 & 8 \\ 3 & 1 & 4 \end{vmatrix} \begin{matrix} 2 & -1 \\ -4 & -3 \\ 3 & 1 \end{matrix} = -24 - 24 - 20 + 45 - 16 - 16 = -55$$

Using relation (D.31) we obtain

$$x_1 = \frac{D_1}{\Delta} = \frac{85}{35} = \frac{17}{7} \qquad x_2 = \frac{D_2}{\Delta} = -\frac{170}{35} = -\frac{34}{7} \qquad x_3 = \frac{D_3}{\Delta} = -\frac{55}{35} = -\frac{11}{7} \tag{D.34}$$

We will verify with MATLAB as follows:

```
% The following code will compute and display the values of v1, v2 and v3.
format rat                          % Express answers in ratio form
B=[2 –1 3; –4 –3 –2; 3 1 –1];       % The elements of the determinant D of matrix B
delta=det(B);                       % Compute the determinant D of matrix B
d1=[5 –1 3; 8 –3 –2; 4 1 –1];       % The elements of D1
detd1=det(d1);                      % Compute the determinant of D1
d2=[2 5 3; –4 8 –2; 3 4 –1];        % The elements of D2
detd2=det(d2);                      % Compute the determinant of D2
d3=[2 –1 5; –4 –3 8; 3 1 4];        % The elements of D3
detd3=det(d3);                      % Compute he determinant of D3
v1=detd1/delta;                     % Compute the value of v1
v2=detd2/delta;                     % Compute the value of v2
v3=detd3/delta;                     % Compute the value of v3
                                    %
disp('v1=');disp(v1);               % Display the value of v1
disp('v2=');disp(v2);               % Display the value of v2
disp('v3=');disp(v3);               % Display the value of v3
```

```
v1=
     17/7
v2=
    -34/7
v3=
    -11/7
```

These are the same values as in (D.34)

D.7 Gaussian Elimination Method

We can find the unknowns in a system of two or more equations also by the *Gaussian elimination method*. With this method, the objective is to eliminate one unknown at a time. This can be done by multiplying the terms of any of the equations of the system by a number such that we can add (or subtract) this equation to another equation in the system so that one of the unknowns will be eliminated. Then, by substitution to another equation with two unknowns, we can find the second unknown. Subsequently, substitution of the two values found can be made into an equation with three unknowns from which we can find the value of the third unknown. This procedure is repeated until all unknowns are found. This method is best illustrated with the following example which consists of the same equations as the previous example.

Example D.11

Use the Gaussian elimination method to find v_1, v_2, and v_3 of the system of equations

$$\begin{aligned} 2v_1 - v_2 + 3v_3 &= 5 \\ -4v_1 - 3v_2 - 2v_3 &= 8 \\ 3v_1 + v_2 - v_3 &= 4 \end{aligned} \tag{D.35}$$

Solution:

As a first step, we add the first equation of (D.35) with the third to eliminate the unknown v_2 and we obtain the equation

$$5v_1 + 2v_3 = 9 \tag{D.36}$$

Next, we multiply the third equation of (D.35) by 3, and we add it with the second to eliminate v_2, and we obtain the equation

$$5v_1 - 5v_3 = 20 \tag{D.37}$$

Subtraction of (D.37) from (D.36) yields

$$7v_3 = -11 \quad \text{or} \quad v_3 = -\frac{11}{7} \tag{D.38}$$

Now, we can find the unknown v_1 from either (D.36) or (D.37). By substitution of (D.38) into (D.36) we obtain

$$5v_1 + 2 \cdot \left(-\frac{11}{7}\right) = 9 \quad \text{or} \quad v_1 = \frac{17}{7} \tag{D.39}$$

Finally, we can find the last unknown v_2 from any of the three equations of (D.35). By substitution into the first equation we obtain

$$v_2 = 2v_1 + 3v_3 - 5 = \frac{34}{7} - \frac{33}{7} - \frac{35}{7} = -\frac{34}{7} \tag{D.40}$$

These are the same values as those we found in Example D.10.

The Gaussian elimination method works well if the coefficients of the unknowns are small integers, as in Example D.11. However, it becomes impractical if the coefficients are large or fractional numbers.

D.8 The Adjoint of a Matrix

Let us assume that A is an *n* square matrix and α_{ij} is the cofactor of a_{ij}. Then *the adjoint of* A, denoted as adjA, is defined as the *n* square matrix below.

$$\text{adjA} = \begin{bmatrix} \alpha_{11} & \alpha_{21} & \alpha_{31} & \cdots & \alpha_{n1} \\ \alpha_{12} & \alpha_{22} & \alpha_{32} & \cdots & \alpha_{n2} \\ \alpha_{13} & \alpha_{23} & \alpha_{33} & \cdots & \alpha_{n3} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_{1n} & \alpha_{2n} & \alpha_{3n} & \cdots & \alpha_{nn} \end{bmatrix} \tag{D.41}$$

We observe that the cofactors of the elements of the ith row (column) of A are the elements of the ith column (row) of adjA.

Example D.12

Compute adjA if Matrix A is defined as

$$\text{A} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 4 \\ 1 & 4 & 3 \end{bmatrix} \tag{D.42}$$

Solution:

$$\text{adjA} = \begin{bmatrix} \begin{bmatrix} 3 & 4 \\ 4 & 3 \end{bmatrix} & -\begin{bmatrix} 2 & 3 \\ 4 & 3 \end{bmatrix} & \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix} \\ -\begin{bmatrix} 1 & 4 \\ 1 & 3 \end{bmatrix} & \begin{bmatrix} 1 & 3 \\ 1 & 3 \end{bmatrix} & -\begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix} \\ \begin{bmatrix} 1 & 3 \\ 1 & 4 \end{bmatrix} & -\begin{bmatrix} 1 & 2 \\ 1 & 4 \end{bmatrix} & \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -7 & 6 & -1 \\ 1 & 0 & -1 \\ 1 & -2 & 1 \end{bmatrix}$$

D.9 Singular and Non–Singular Matrices

An n square matrix A is called *singular* if detA = 0; if detA ≠ 0, A is called *non–singular.*

Example D.13

Matrix A is defined as

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 5 & 7 \end{bmatrix} \tag{D.43}$$

Determine whether this matrix is singular or non–singular.

Solution:

$$\det A = \begin{vmatrix} 1 & 2 & 3 & 1 & 2 \\ 2 & 3 & 4 & 2 & 3 \\ 3 & 5 & 7 & 3 & 5 \end{vmatrix} = 21 + 24 + 30 - 27 - 20 - 28 = 0$$

Therefore, matrix A is singular.

D.10 The Inverse of a Matrix

If A and B are n square matrices such that $AB = BA = I$, where I is the identity matrix, B is called the *inverse* of A, denoted as $B = A^{-1}$, and likewise, A is called the inverse of B, that is, $A = B^{-1}$

If a matrix A is non-singular, we can compute its inverse A^{-1} from the relation

$$\boxed{A^{-1} = \frac{1}{\det A}\text{adj}A} \tag{D.44}$$

Example D.14

Matrix A is defined as

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 4 \\ 1 & 4 & 3 \end{bmatrix} \tag{D.45}$$

Compute its inverse, that is, find A^{-1}

Solution:

Here, $\det A = 9 + 8 + 12 - 9 - 16 - 6 = -2$, and since this is a non-zero value, it is possible to compute the inverse of A using (D.44).

From Example D.12,

$$\text{adj}A = \begin{bmatrix} -7 & 6 & -1 \\ 1 & 0 & -1 \\ 1 & -2 & 1 \end{bmatrix}$$

Then,

$$A^{-1} = \frac{1}{\det A}\text{adj}A = \frac{1}{-2}\begin{bmatrix} -7 & 6 & -1 \\ 1 & 0 & -1 \\ 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 3.5 & -3 & 0.5 \\ -0.5 & 0 & 0.5 \\ -0.5 & 1 & -0.5 \end{bmatrix} \tag{D.46}$$

Check with MATLAB:

```
A=[1  2  3;  1  3  4;  1  4  3],  invA=inv(A)      % Define matrix A and compute its inverse

A =
     1     2     3
     1     3     4
     1     4     3

invA =
    3.5000   -3.0000    0.5000
   -0.5000         0    0.5000
   -0.5000    1.0000   -0.5000
```

Multiplication of a matrix A by its inverse A^{-1} produces the identity matrix I, that is,

$$AA^{-1} = I \quad \text{or} \quad A^{-1}A = I \tag{D.47}$$

Example D.15

Prove the validity of (D.47) for the Matrix A defined as

$$A = \begin{bmatrix} 4 & 3 \\ 2 & 2 \end{bmatrix}$$

Proof:

$$\det A = 8 - 6 = 2 \quad \text{and} \quad \text{adj}A = \begin{bmatrix} 2 & -3 \\ -2 & 4 \end{bmatrix}$$

Then,

$$A^{-1} = \frac{1}{\det A}\text{adj}A = \frac{1}{2}\begin{bmatrix} 2 & -3 \\ -2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & -3/2 \\ -1 & 2 \end{bmatrix}$$

and

$$AA^{-1} = \begin{bmatrix} 4 & 3 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 1 & -3/2 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 4-3 & -6+6 \\ 2-2 & -3+4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

D.11 Solution of Simultaneous Equations with Matrices

Consider the relation

$$AX = B \tag{D.48}$$

where A and B are matrices whose elements are known, and X is a matrix (a column vector) whose elements are the unknowns. We assume that A and X are conformable for multiplication.

Multiplication of both sides of (D.48) by A^{-1} yields:

$$A^{-1}AX = A^{-1}B = IX = A^{-1}B \tag{D.49}$$

or

$$\boxed{X=A^{-1}B} \tag{D.50}$$

Therefore, we can use (D.50) to solve any set of simultaneous equations that have solutions. We will refer to this method as the *inverse matrix method of solution* of simultaneous equations.

Example D.16

For the system of the equations

$$\left.\begin{aligned} 2x_1 + 3x_2 + x_3 &= 9 \\ x_1 + 2x_2 + 3x_3 &= 6 \\ 3x_1 + x_2 + 2x_3 &= 8 \end{aligned}\right\} \tag{D.51}$$

compute the unknowns x_1, x_2, and x_3 using the inverse matrix method.

Solution:

In matrix form, the given set of equations is $AX = B$ where

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad B = \begin{bmatrix} 9 \\ 6 \\ 8 \end{bmatrix} \tag{D.52}$$

Then,

$$X = A^{-1}B \tag{D.53}$$

or

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 9 \\ 6 \\ 8 \end{bmatrix} \tag{D.54}$$

Next, we find the determinant $\mathrm{det}A$, and the adjoint $\mathrm{adj}A$.

$$\mathrm{det}A = 18 \quad \text{and} \quad \mathrm{adj}A = \begin{bmatrix} 1 & -5 & 7 \\ 7 & 1 & -5 \\ -5 & 7 & 1 \end{bmatrix}$$

Therefore,

$$A^{-1} = \frac{1}{\mathrm{det}A}\,\mathrm{adj}A = \frac{1}{18}\begin{bmatrix} 1 & -5 & 7 \\ 7 & 1 & -5 \\ -5 & 7 & 1 \end{bmatrix}$$

and with relation (D.53) we obtain the solution as follows:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{18}\begin{bmatrix} 1 & -5 & 7 \\ 7 & 1 & -5 \\ -5 & 7 & 1 \end{bmatrix}\begin{bmatrix} 9 \\ 6 \\ 8 \end{bmatrix} = \frac{1}{18}\begin{bmatrix} 35 \\ 29 \\ 5 \end{bmatrix} = \begin{bmatrix} 35/18 \\ 29/18 \\ 5/18 \end{bmatrix} = \begin{bmatrix} 1.94 \\ 1.61 \\ 0.28 \end{bmatrix} \tag{D.55}$$

To verify our results, we could use the MATLAB's **inv(A)** function, and then multiply A^{-1} by B. However, it is easier to use the *matrix left division* operation $X = A \backslash B$; this is MATLAB's solution of $A^{-1}B$ for the matrix equation $A \cdot X = B$, where matrix X is the same size as matrix B. For this example,

```
A=[2  3  1; 1  2  3; 3  1  2]; B=[9  6  8]'; X=A \ B

  X =
       1.9444
       1.6111
       0.2778
```

Example D.17

For the electric circuit of Figure D.1,

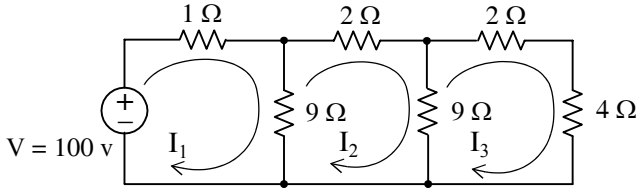


Figure D.1. Electric circuit for Example D.17

the loop equations are

$$\begin{aligned} 10I_1 - 9I_2 &= 100 \\ -9I_1 + 20I_2 - 9I_3 &= 0 \\ -9I_2 + 15I_3 &= 0 \end{aligned} \tag{D.56}$$

Use the inverse matrix method to compute the values of the currents I_1, I_2, and I_3

Solution:

For this example, the matrix equation is $RI = V$ or $I = R^{-1}V$, where

$$R = \begin{bmatrix} 10 & -9 & 0 \\ -9 & 20 & -9 \\ 0 & -9 & 15 \end{bmatrix}, \quad V = \begin{bmatrix} 100 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad I = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}$$

The next step is to find R^{-1}. It is found from the relation

$$R^{-1} = \frac{1}{\det R}\,\mathrm{adj}R \tag{D.57}$$

Therefore, we must find the determinant and the adjoint of R. For this example, we find that

$$\text{detR} = 975, \quad \text{adjR} = \begin{bmatrix} 219 & 135 & 81 \\ 135 & 150 & 90 \\ 81 & 90 & 119 \end{bmatrix} \tag{D.58}$$

Then,

$$R^{-1} = \frac{1}{\text{detR}}\text{adjR} = \frac{1}{975}\begin{bmatrix} 219 & 135 & 81 \\ 135 & 150 & 90 \\ 81 & 90 & 119 \end{bmatrix}$$

and

$$I = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \frac{1}{975}\begin{bmatrix} 219 & 135 & 81 \\ 135 & 150 & 90 \\ 81 & 90 & 119 \end{bmatrix}\begin{bmatrix} 100 \\ 0 \\ 0 \end{bmatrix} = \frac{100}{975}\begin{bmatrix} 219 \\ 135 \\ 81 \end{bmatrix} = \begin{bmatrix} 22.46 \\ 13.85 \\ 8.31 \end{bmatrix}$$

Check with MATLAB:

```
R=[10 –9  0; –9  20 –9; 0 –9 15]; V=[100 0 0]'; I=R\V; fprintf(' \n');...
fprintf('I1 = %4.2f \t', I(1)); fprintf('I2 = %4.2f \t', I(2)); fprintf('I3 = %4.2f \t', I(3)); fprintf(' \n')
```

```
  I1 = 22.46    I2 = 13.85    I3 = 8.31
```

We can also use subscripts to address the individual elements of the matrix. Accordingly, the MATLAB script above could also have been written as:

```
R(1,1)=10; R(1,2)=–9;     % No need to make entry for A(1,3) since it is zero.
R(2,1)=–9; R(2,2)=20; R(2,3)=–9; R(3,2)=–9; R(3,3)=15; V=[100 0 0]'; I=R\V; fprintf(' \n');...
fprintf('I1 = %4.2f \t', I(1)); fprintf('I2 = %4.2f \t', I(2)); fprintf('I3 = %4.2f \t', I(3)); fprintf(' \n')
```

```
  I1 = 22.46    I2 = 13.85    I3 = 8.31
```

Spreadsheets also have the capability of solving simultaneous equations with real coefficients using the inverse matrix method. For instance, we can use Microsoft Excel's MINVERSE (Matrix Inversion) and MMULT (Matrix Multiplication) functions, to obtain the values of the three currents in Example D.17.

The procedure is as follows:

1. We begin with a blank spreadsheet and in a block of cells, say B3:D5, we enter the elements of matrix *R* as shown in Figure D.2. Then, we enter the elements of matrix V in G3:G5.
2. Next, we compute and display the inverse of R, that is, R^{-1}. We choose B7:D9 for the elements of this inverted matrix. We format this block for number display with three decimal

places. With this range highlighted and making sure that the cell marker is in B7, we type the formula

```
=MININVERSE(B3:D5)
```

and we press the *Crtl-Shift-Enter* keys simultaneously. We observe that $\mathbf{R}^{-1}$ appears in these cells.

3. Now, we choose the block of cells G7:G9 for the values of the current $\mathbf{I}$. As before, we highlight them, and with the cell marker positioned in G7, we type the formula

```
=MMULT(B7:D9,G3:G5)
```

and we press the Crtl-Shift-Enter keys simultaneously. The values of $\mathbf{I}$ then appear in G7:G9.

	A	B	C	D	E	F	G	H
1	Spreadsheet for Matrix Inversion and Matrix Multiplication							
2								
3		10	-9	0			100	
4	R=	-9	20	-9		V=	0	
5		0	-9	15			0	
6								
7		0.225	0.138	0.083			22.462	
8	R^{-1}=	0.138	0.154	0.092		I=	13.846	
9		0.083	0.092	0.122			8.3077	
10								

Figure D.2. Solution of Example D.17 with a spreadsheet

Example D.18

For the phasor circuit of Figure D.18

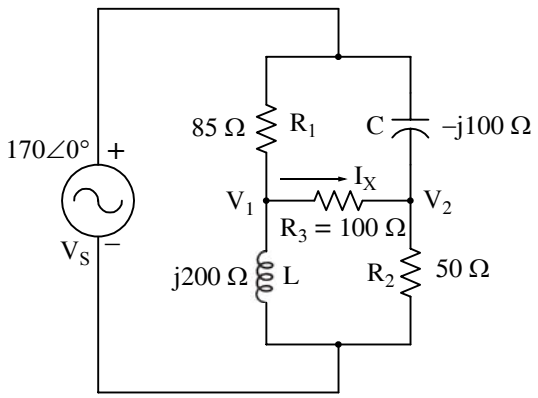


Figure D.3. Circuit for Example D.18

the current I_X can be found from the relation

$$I_X = \frac{V_1 - V_2}{R_3} \tag{D.59}$$

and the voltages V_1 and V_2 can be computed from the nodal equations

$$\frac{V_1 - 170\angle 0°}{85} + \frac{V_1 - V_2}{100} + \frac{V_1 - 0}{j200} = 0 \tag{D.60}$$

and

$$\frac{V_2 - 170\angle 0°}{-j100} + \frac{V_2 - V_1}{100} + \frac{V_2 - 0}{50} = 0 \tag{D.61}$$

Compute, and express the current I_x in both rectangular and polar forms by first simplifying like terms, collecting, and then writing the above relations in matrix form as YV = I, where Y = Admittance, V = Voltage, and I = Current

Solution:

The Y matrix elements are the coefficients of V_1 and V_2. Simplifying and rearranging the nodal equations of (D.60) and (D.61), we obtain

$$\begin{aligned}(0.0218 - j0.005)V_1 - 0.01V_2 &= 2\\ -0.01V_1 + (0.03 + j0.01)V_2 &= j1.7\end{aligned} \tag{D.62}$$

Next, we write (D.62) in matrix form as

$$\underbrace{\begin{bmatrix}0.0218 - j0.005 & -0.01\\ -0.01 & 0.03 + j0.01\end{bmatrix}}_{Y}\underbrace{\begin{bmatrix}V_1\\ V_2\end{bmatrix}}_{V} = \underbrace{\begin{bmatrix}2\\ j1.7\end{bmatrix}}_{I} \tag{D.63}$$

where the matrices Y, V, and I are as indicated.

We will use MATLAB to compute the voltages V_1 and V_2, and to do all other computations. The script is shown below.

```
Y=[0.0218–0.005j  –0.01;  –0.01  0.03+0.01j]; I=[2; 1.7j]; V=Y\I;  % Define Y, I, and find V
fprintf('\n');                                                    % Insert a line
disp('V1 = '); disp(V(1)); disp('V2 = '); disp(V(2));             % Display values of V1 and V2
```

```
V1 =
  1.0490e+002 + 4.9448e+001i
```

```
V2 =
  53.4162 + 55.3439i
```

Next, we find I_X from

```
R3=100; IX=(V(1)–V(2))/R3                    % Compute the value of I_X
```

```
IX =
   0.5149 - 0.0590i
```

This is the rectangular form of I_X. For the polar form we use the MATLAB script

```
magIX=abs(IX), thetaIX=angle(IX)*180/pi  % Compute the magnitude and the angle in degrees
```

```
magIX =
    0.5183

thetaIX =
   -6.5326
```

Therefore, in polar form

$$I_X = 0.518\angle{-6.53°}$$

Spreadsheets have limited capabilities with complex numbers, and thus we cannot use them to compute matrices that include complex numbers in their elements as in Example D.18.

Appendix E

Window Functions

This appendix is an introduction to window functions. We discuss the rectangular, triangular, Hanning, Hamming, Blackman, and Kaiser windows. An example using each is provided for illustration of their uses.

E.1 Window Function Defined

A *window function* is a function that is zero–valued outside of some chosen interval. For instance, a function that is constant inside the interval and zero elsewhere is called a *rectangular window,* and describes the shape of its graphical representation. When another function or a signal (data) is multiplied by a window function, the product is also zero–valued outside the interval: all that is left is the "view" through the window. Applications of window functions include spectral analysis, and filter design.

When selecting an appropriate window function for an application, a comparison graph may be useful. The most important parameter is usually the stop band attenuation close to the main lobe.

All of the window functions that we will discuss are even functions of time when centered at the origin.

E.2 Common Window Functions

Based on the discussion in the previous section, it appears that a rectangular function would be the ideal window function to terminate an impulse response with an infinite number of terms. For instance, let us assume that the impulse response `h[n]` shown in Figure E.1(a) below converges uniformly and is represented by a portion of the amplitude response `A(f)` shown in Figure E.1(b).

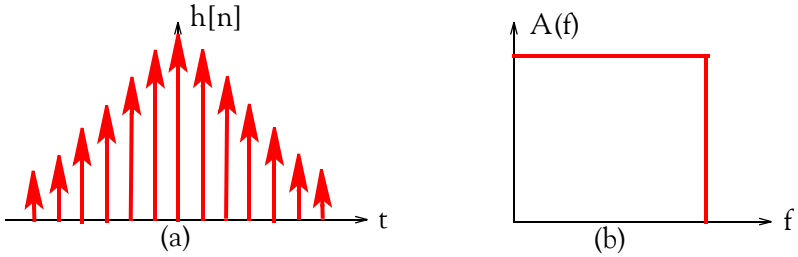


Figure E.1. Impulse response with an infinite number of terms

Next, let us assume that the impulse response `h[n]` is terminated abruptly without changing any of its coefficients, as shown in Figure E.2(a). In this case, the resulting amplitude response `A'(f)` will be subject to undesired oscillations and poor convergence as shown in Figure E.2(b).

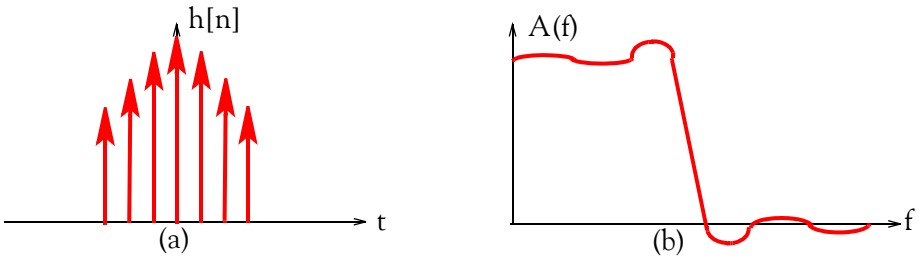


Figure E.2. Resultant amplitude response when the impulse response is abruptly truncated

Therefore, we are seeking a suitable window function that will result in an amplitude response with a lower ripple in the stop band such as the one shown in Figure E.3.

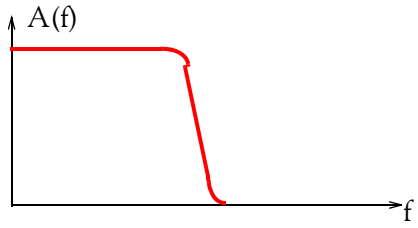


Figure E.3. Acceptable amplitude response

Although an impulse response may converge uniformly, when it is multiplied* by a rectangular window function it results in an undesirable amplitude response. However, the rectangular window function is the basis in studying window functions, so we begin with the description of the rectangular window function.

E.2.1 Rectangular Window Function

The rectangular window function is defined as

$$\begin{aligned} f(t)_{rect} &= 1 \quad \text{for } |t| < \frac{\tau}{2} \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{E.1}$$

and its time–domain waveform is as shown in Figure E.4.

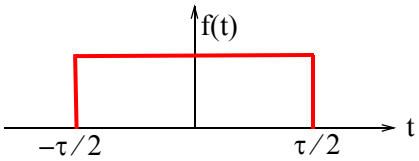


Figure E.4. Rectangular window function in time–domain

* *We recall that multiplication in the time domain corresponds to convolution in the frequency domain.*

The Fourier transform of the rectangular window function for $\tau = 1$ is

$$F(\omega)_{rect} = \frac{\sin(\pi\omega/2)}{\pi\omega/2} \tag{E.2}$$

It is customary and convenient to express the amplitude of $F(\omega)$ in a decibel scale. Then

$$F(\omega)_{rect(dB)} = 20\log\frac{|F(\omega)|}{F(0)} \tag{E.3}$$

and relation (F.3) is normalized with respect to the DC value $F(0)$. A typical amplitude response of the rectangular window function is shown in Figure E.5, and it was created with the MATLAB script below.

```
fplot('20.*log10(abs(sin(pi.*x)./(pi.*x)))',[0 5 5 –50])
```

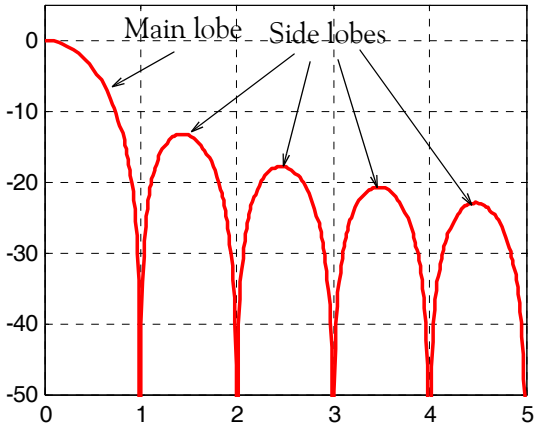


Figure E.5. Typical amplitude response for the rectangular window function

The amplitude response in Figure E.5 consists of a main lobe at the middle* of the spectrum and side lobes located on either side (positive and negative) of the main lobe.

It is desired that a window function should have a narrow main lobe and the maximum side lobe level should be very small in relation to the main lobe. Unfortunately, both of these requirements cannot be optimized simultaneously, and thus we must decide on a suitable compromise between these two requirements.

Figure E.6 shows the normalized frequency domain plot for the rectangular window function created with the rectwin MATLAB function, and the script below.

```
n=50; w=rectwin(n); [W,f]=freqz(w/sum(w),1,512,2); plot(f,20*log10(abs(W)));grid
```

* *Generally, only the positive half of the response is shown in all window functions.*

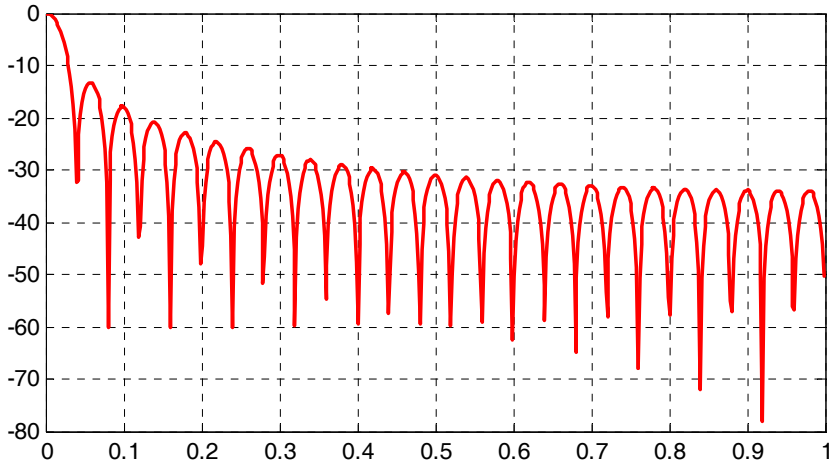


Figure E.6. Normalized frequency domain plot for the rectangular window function created with MATLAB

We can also use the MATLAB **Window Visualization Tool**. The script below generates the plot shown in Figure E.7.

```
L=50; wvtool(rectwin(L))
```

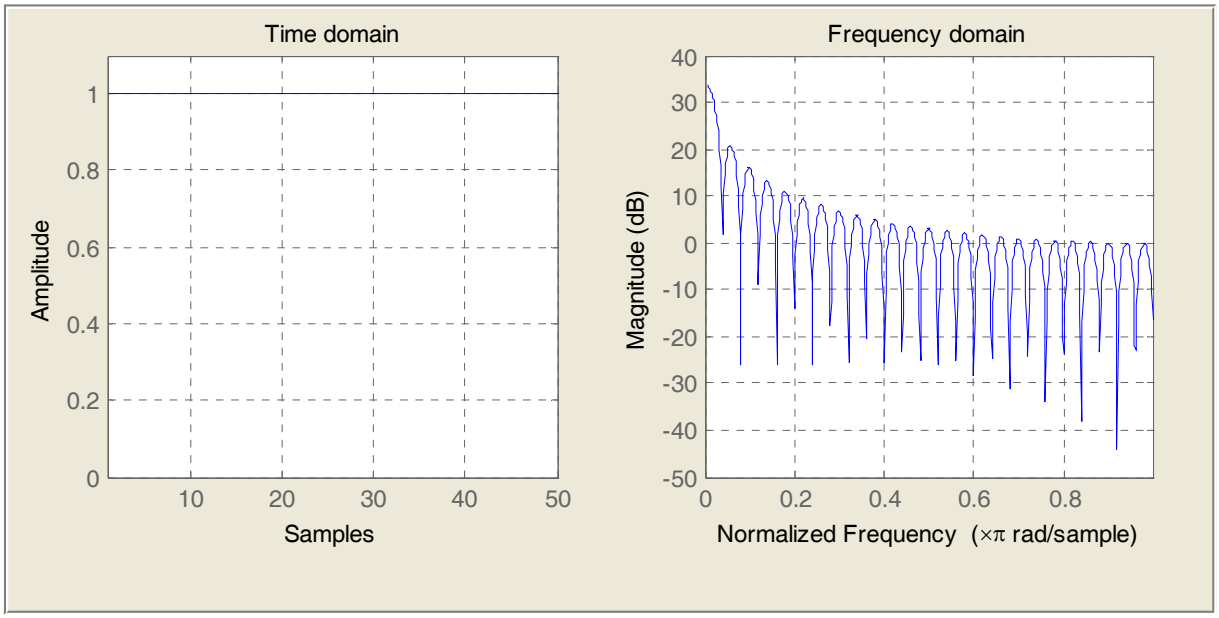


Figure E.7. Rectangular window function generated with the MATLAB Window Visualization Tool

E.2.2 Triangular Window Function

The triangular window function is defined as

$$f(t)_{triang} = 1 - \frac{2|t|}{\tau} \quad \text{for } |t| < \frac{\tau}{2}$$
$$= 0 \quad \text{otherwise} \tag{E.4}$$

and its time–domain waveform is as shown in Figure E.8.

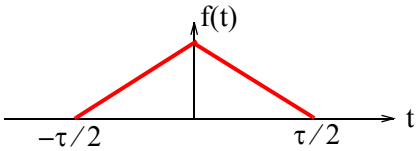


Figure E.8. Triangular window function in time–domain

The Fourier transform of the triangular window function for $\tau = 1$ is

$$F(\omega)_{triang} = \frac{1}{2}\left(\frac{\sin(\pi\omega/2)}{\pi\omega/2}\right)^2 \tag{E.5}$$

A typical amplitude response of the triangular window function is shown in Figure E.9 and it was created with the MATLAB script below.

```
fplot('20.*log10(abs(sin(0.5.*pi.*x)./(0.5.*pi.*x)).^2)',[0 5 5 –50])
```

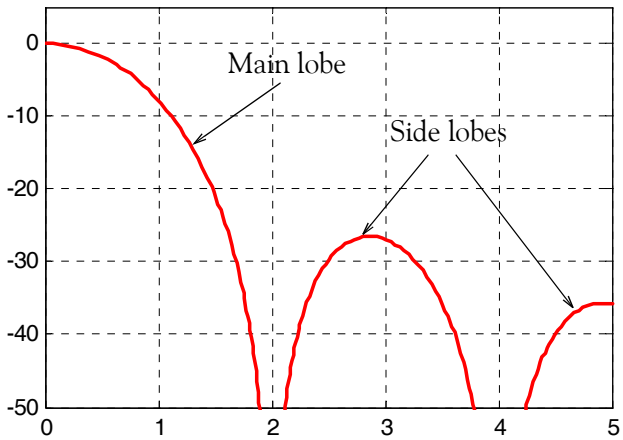


Figure E.9. Typical amplitude response for the triangular window function

We observe that the width of the main lobe of the triangular window function is about twice as wide as that of the rectangular window function, but the first side lobe is much lower.

Figure E.10 shows the normalized frequency domain plot for the triangular window function created with the triang MATLAB function, and the script below.

```
n=50; w=triang(n); [W,f]=freqz(w/sum(w),1,512,2); plot(f,20*log10(abs(W)));grid
```

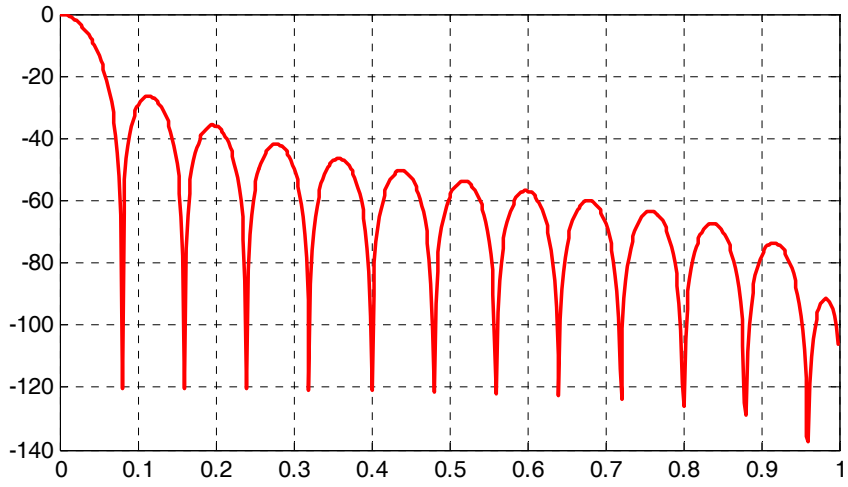


Figure E.10. Normalized frequency domain plot for the triangular window function created with MATLAB

We can also use the MATLAB Window Visualization Tool. The script below generates the plot shown in Figure E.11.

```
L=50; wvtool(triang(L))
```

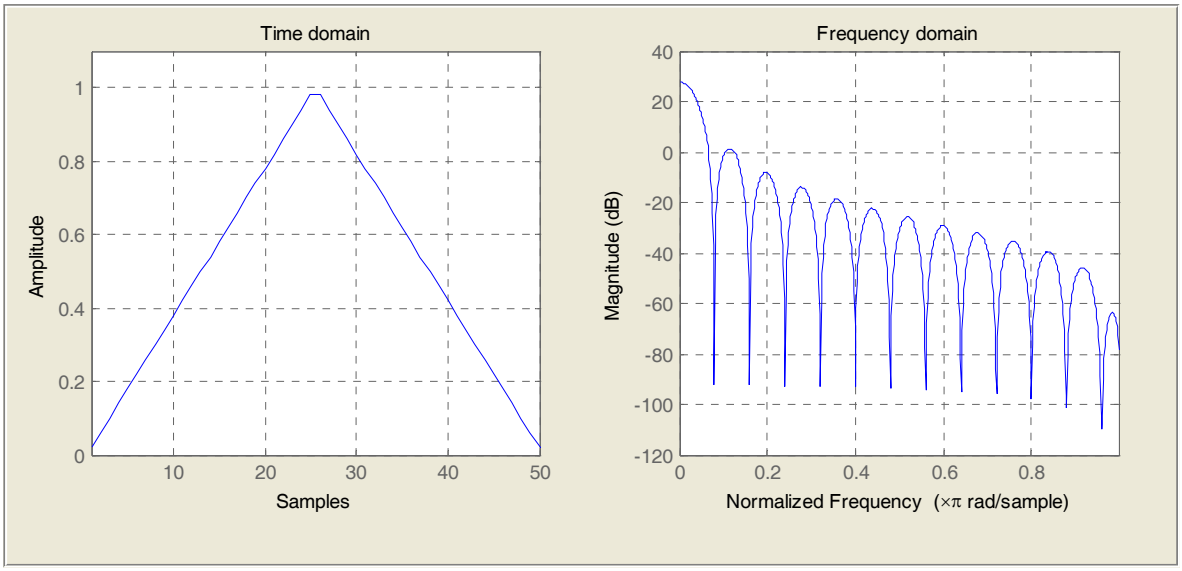


Figure E.11. Triangular window function generated with the MATLAB Window Visualization Tool

E.2.3 Hanning Window Function

The Hanning* window function is defined as

$$f(t)_{Hann} = \cos^2(\pi t/\tau) = \frac{1}{2}\left(1 + \cos\frac{2\pi t}{\tau}\right) \quad \text{for } |t| < \frac{\tau}{2}$$
$$= 0 \quad \text{otherwise} \tag{E.6}$$

and its time–domain waveform is as shown in Figure E.12.

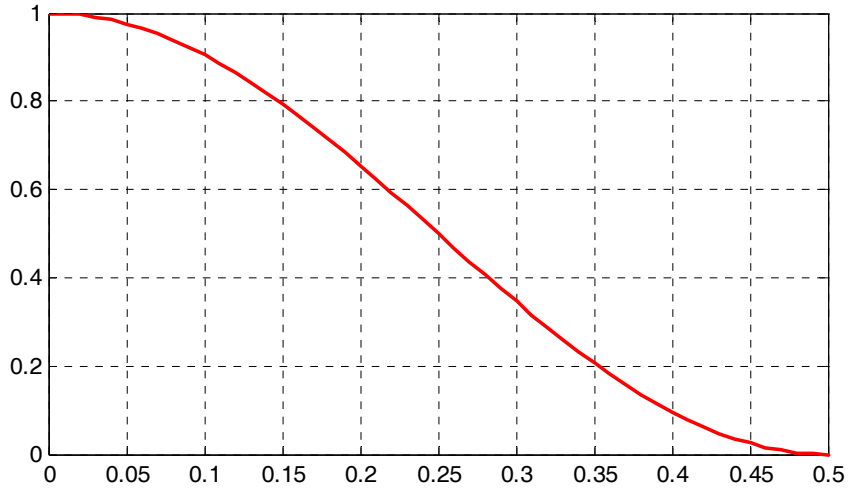


Figure E.12. Hanning window function in time–domain

The Fourier transform of the Hanning window function for $\tau = 1$ is

$$F(\omega)_{Hann} = \frac{1}{2}\left(\frac{\sin(\pi\omega)}{\pi\omega}\right)\left(\frac{1}{1-\omega^2}\right) \tag{E.7}$$

A typical amplitude response of the Hanning window function is shown in Figure E.13 and it was created with the MATLAB script below.

```
fplot('20.*log10(abs(sin(pi.*x)./(pi.*x)).*(1./(1–(x+eps).^2)))',[0 5 –50 0])
```

* *This window function is also known as Hann window function or cosine–squared window function.*

Figure E.13. Typical amplitude response for the Hanning window function

Figure E.14 shows the normalized frequency domain plot for the Hanning window function created with the hann MATLAB function, and the script below.

```
n=50; w=hann(n); [W,f]=freqz(w/sum(w),1,512,2); plot(f,20*log10(abs(W)));grid
```

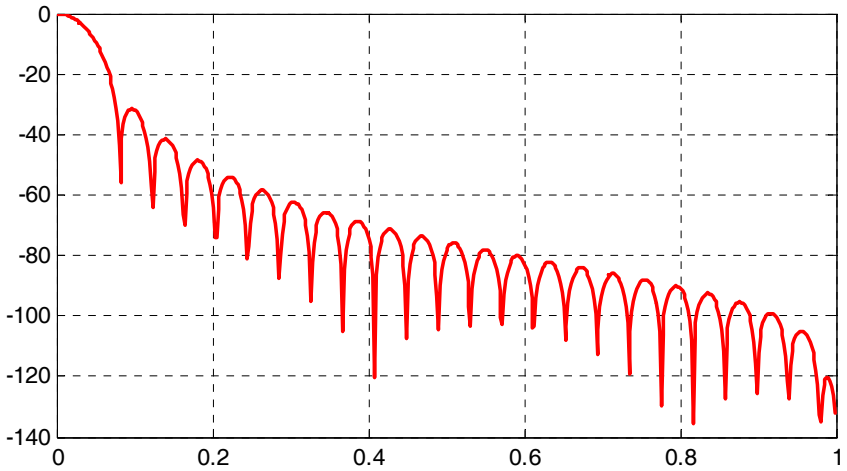


Figure E.14. Normalized frequency domain plot for the Hanning window function created with MATLAB

We can also use the MATLAB Window Visualization Tool. The script below generates the plot shown in Figure E.15.

```
L=50; wvtool(hann(L))
```

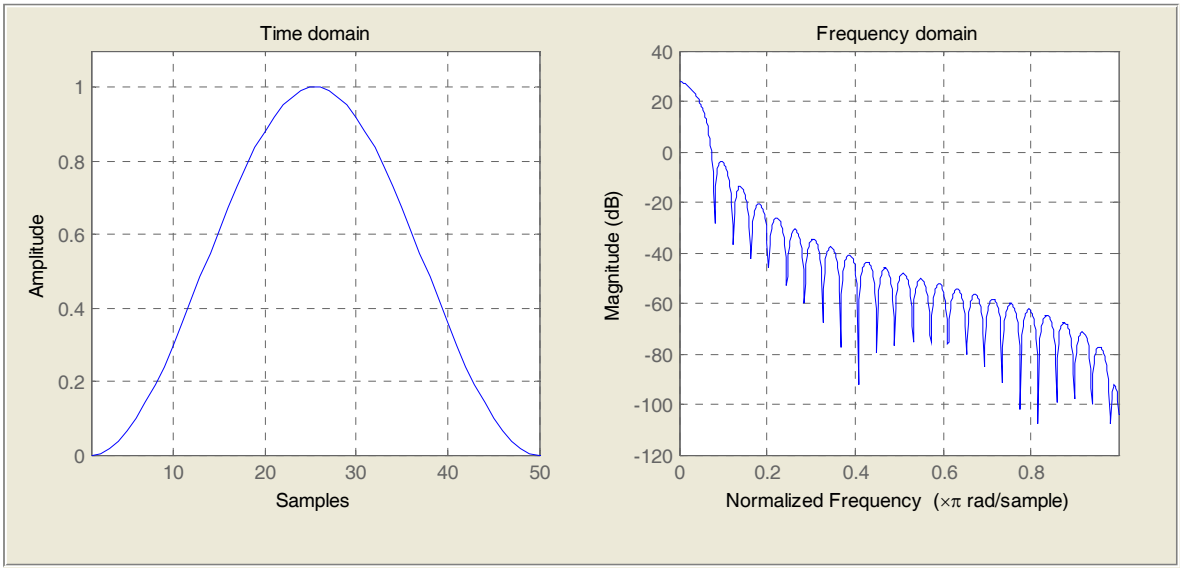


Figure E.15. Hanning window function generated with the MATLAB Window Visualization Tool

E.2.4 Hamming Window Function

The Hamming window function is defined as

$$f(t)_{Hamm} = 0.54 + 0.46\cos\frac{2\pi t}{\tau} \quad \text{for } |t| < \frac{\tau}{2}$$
$$= 0 \quad \text{otherwise} \tag{E.8}$$

and its time–domain waveform is as shown in Figure E.16.

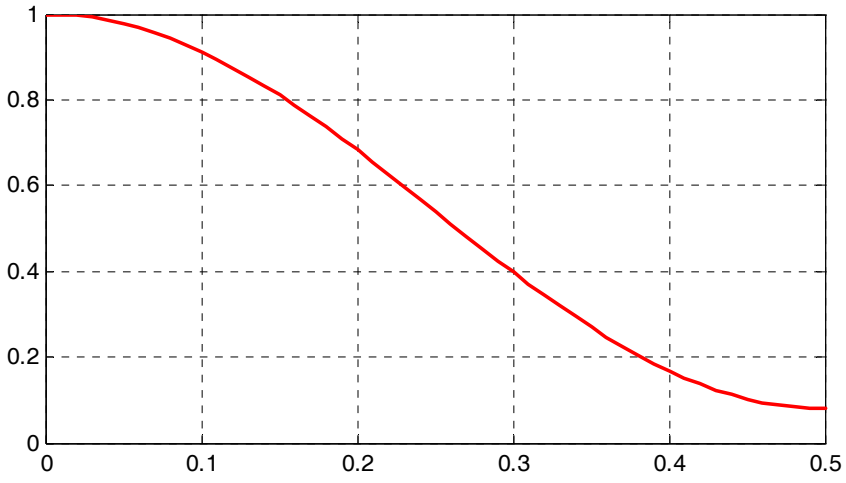


Figure E.16. Hamming window function in time–domain

The Fourier transform of the Hamming window function for $\tau = 1$ is

$$F(\omega)_{\text{Hamm}} = \left(\frac{\sin(\pi\omega)}{\pi\omega}\right)\left(\frac{0.54 - 0.08\omega^2}{1 - \omega^2}\right) \tag{E.9}$$

A typical amplitude response of the Hamming window function is shown in Figure E.17 and it was created with the MATLAB script below.

```
fplot('20.*log10(abs(sin(pi.*x)./(pi.*x+eps)).*((0.54–0.08.*x.^2)./(1–x.^2)))',[0 5 –60 0])
```

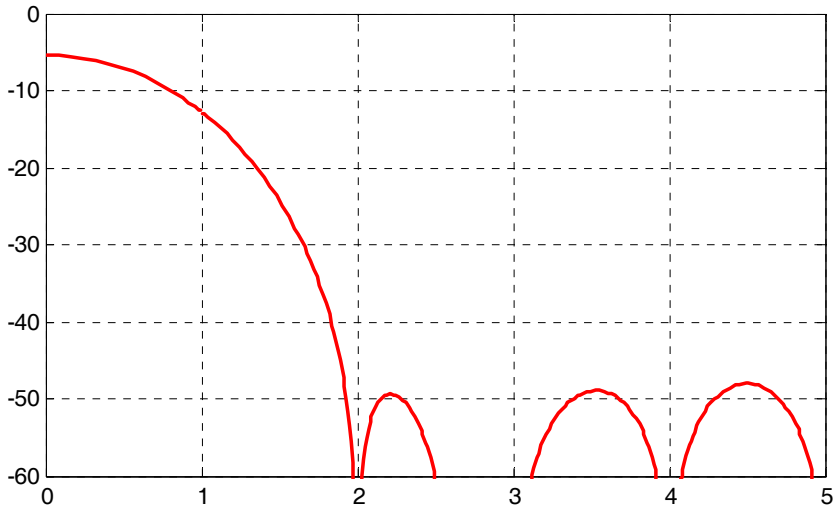


Figure E.17. Typical amplitude response for the Hamming window function

Figure E.18 shows the normalized frequency domain plot for the Hamming window function created with the hamming MATLAB function, and the script below.

```
n=50; w=hamming(n); [W,f]=freqz(w/sum(w),1,512,2); plot(f,20*log10(abs(W)));grid
```

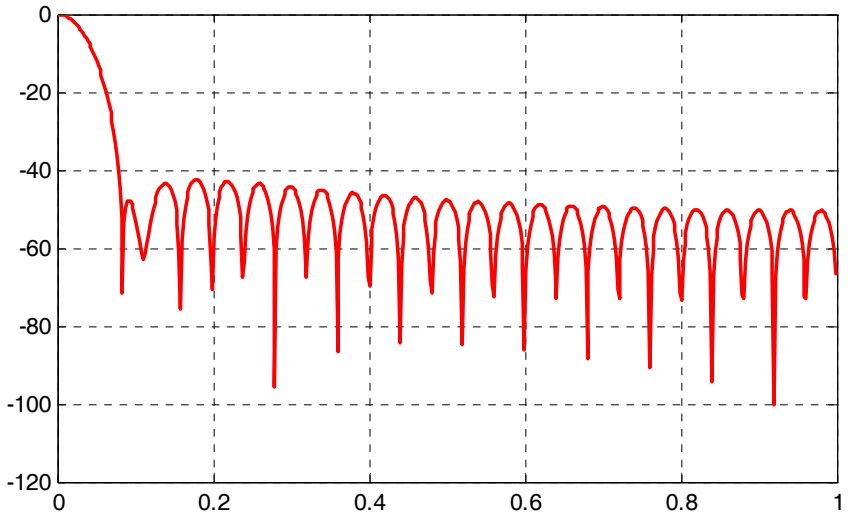


Figure E.18. Normalized frequency domain plot for the Hamming window function created with MATLAB

We can also use the MATLAB Window Visualization Tool. The script below generates the plot shown in Figure E.19.

```
L=50; wvtool(hamming(L))
```

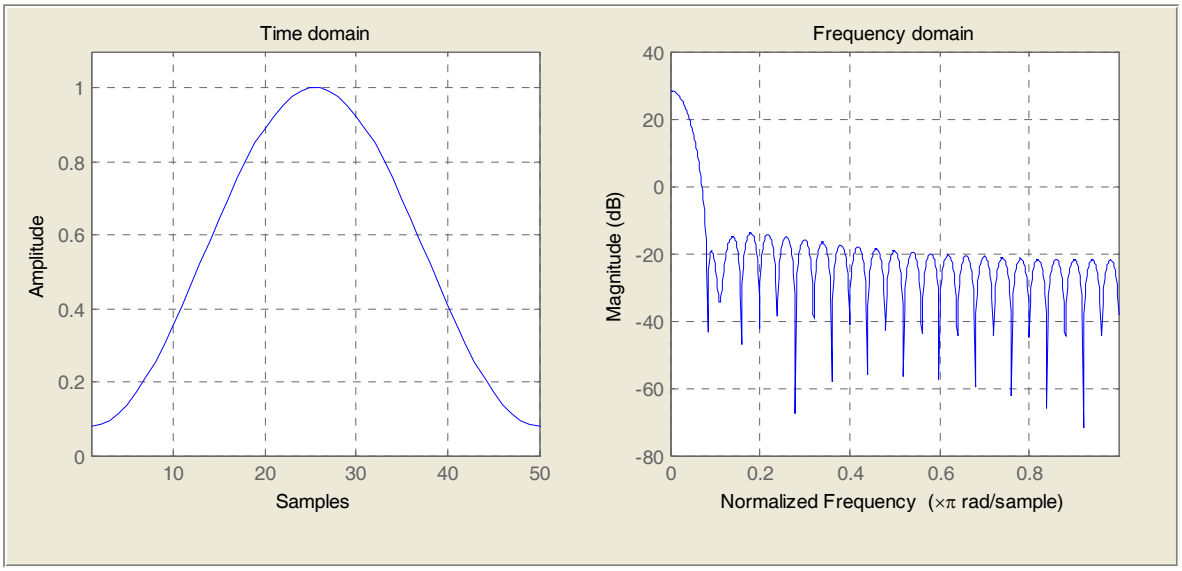


Figure E.19. Hamming window function generated with the MATLAB Window Visualization Tool

E.2.5 Blackman Window Function

The Blackman window function is defined as

$$f(t)_{Black} = 0.42 + 0.5\cos\frac{2\pi t}{\tau} + 0.08\cos\frac{4\pi t}{\tau} \quad \text{for } |t| < \frac{\tau}{2}$$
$$= 0 \quad \text{otherwise} \tag{E.10}$$

and its time–domain waveform is as shown in Figure E.20.

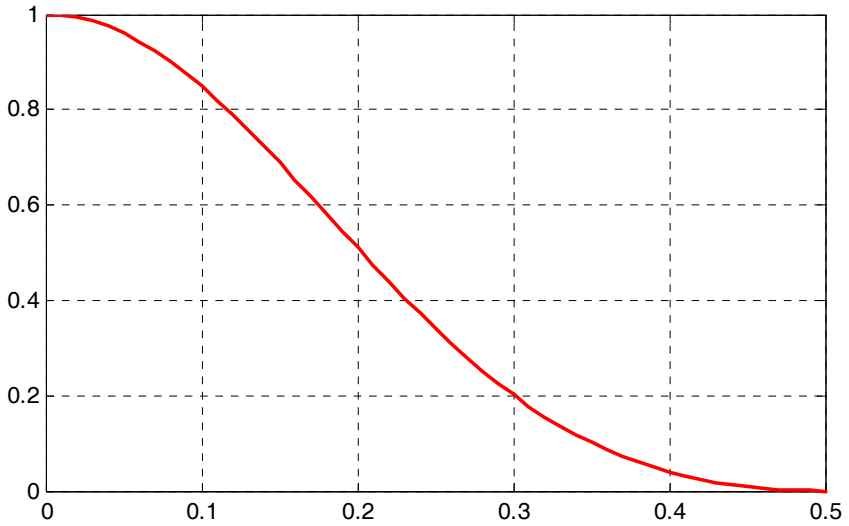


Figure E.20. Blackman window function in time–domain

The Fourier transform of the Blackman window function for $\tau = 1$ is

$$F(\omega)_{Black} = \left(\frac{\sin(\pi\omega)}{\pi\omega}\right)\left(0.42 + \frac{0.5\omega^2}{1-\omega^2} - \frac{0.08\omega^2}{4-\omega^2}\right) \tag{E.11}$$

A typical amplitude response of the Blackman window function is shown in Figure E.21 and it was created with the MATLAB script below.

```
fplot('20.*log10(abs(sin(pi.*x)./(pi.*x)).*((0.54+0.5.*x.^2)./(1–x.^2)–0.08.*x.^2./(4–x.^2)))',[0 5 –50 0])
```

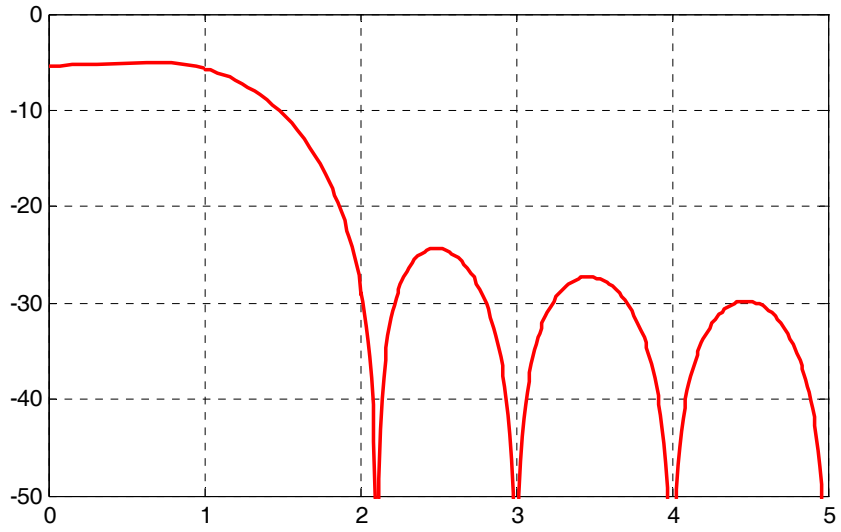


Figure E.21. Typical amplitude response for the Blackman window function

Figure E.22 shows the normalized frequency domain plot for the Blackman window function created with the blackman MATLAB function, and the script below.

```
n=50; w=blackman(n); [W,f]=freqz(w/sum(w),1,512,2); plot(f,20*log10(abs(W)));grid
```

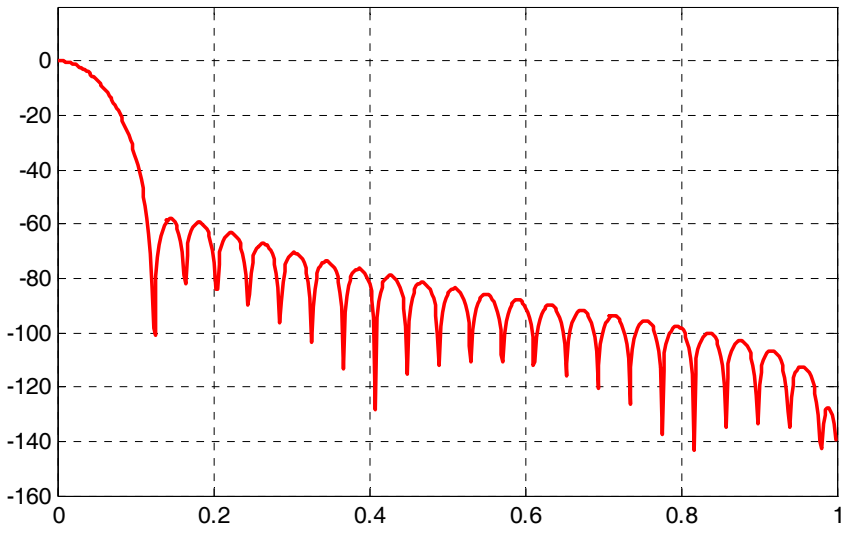


Figure E.22. Normalized frequency domain plot for the Hamming window function created with MATLAB

We can also use the MATLAB Window Visualization Tool. The script below generates the plot shown in Figure E.23.

```
L=50; wvtool(blackman(L))
```

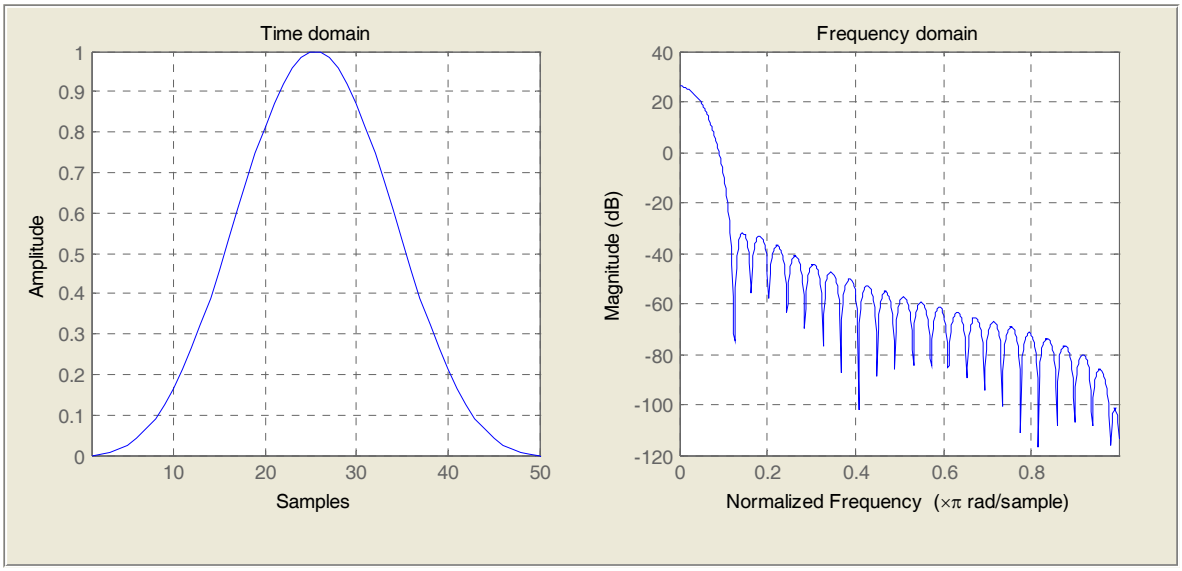


Figure E.23. Blackman window function generated with the MATLAB Window Visualization Tool

E.2.6 Kaiser Family of Window Functions

The Kaiser window function is a family of window functions described by the relation

$$w(n)_{\text{Kaiser}} = \frac{I_0[\beta\sqrt{1-(n-L)^2/L^2}]}{I_0\beta} \qquad \text{for } n=0,1,\ldots,L-1 \tag{E.12}$$

where I_0 is the modified Bessel function* of the first kind and order zero, and it is given as

$$I_0(\beta) = \sum_{k=0}^{\infty}\left(\frac{0.5\beta^k}{k!}\right)^2 \tag{E.13}$$

The series in (E.13) converges after about 20 terms in the summation, and since $I_0(0) = 1$, the Kaiser window has the value $1/I_0(\beta)$ at the end points $n = 0$ and $n = L-1$, and it is symmetric about its middle point $n = L$.

In (E. 12), the variable β is a parameter that can be varied to provide a trade-off between the main lobe width and the side lobe level. Large values of β result in wider main lobe widths and

* *Certain differential equations with variable coefficients resemble the Bessel equation and thus their solutions are referred to as modified Bessel functions. For a discussion on the Bessel equation and Bessel functions, please refer to Numerical Analysis Using* MATLAB *and Excel, ISBN 978–1–934404–03–4.*

smaller side lobe levels.

MATLAB provides the Kaiser window function as

```
w=kaiser(L, beta)
```

and this returns an L–point Kaiser window* in the column vector w.

Figure E. 24 shows the Kaiser functions for $L = 50$ with $\beta = 1$, $\beta = 4$, and $\beta = 9$. These functions were plotted with the MATLAB script below.

```
plot([kaiser(50, 1), kaiser(50,4), kaiser(50,9)]);
```

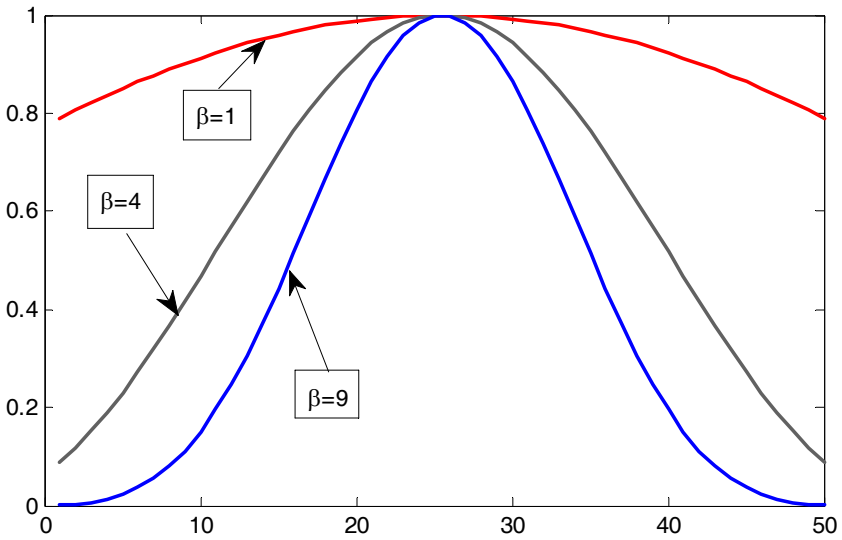


Figure E.24. Kaiser window functions with L=50 and $\beta = 1$, $\beta = 4$, and $\beta = 9$

Figure E.25 shows the corresponding frequency domain plots for the Kaiser window functions in Figure E.24. These plots were created with the MATLAB script below.

```
n=50;
w1 =kaiser(n, 1); w2=kaiser(n,4); w3=kaiser(n,9);
[W1,f]=freqz(w1/sum(w1),1,512,2); [W2,f]=freqz(w2/sum(w2),1,512,2);
[W3,f]=freqz(w3/sum(w3),1,512,2); plot(f,20*log10(abs([W1 W2 W3])));grid;
```

E.3 Other Window Functions

Table E.1 lists the window functions we've discussed in the previous sections, as well as others along with the MATLAB function names and are included in the MATLAB Signal Processing Toolbox.

* *When the expression under the radical in* (E.*12*) *is negative, the function can be expressed in terms of the hyperbolic sine function.*

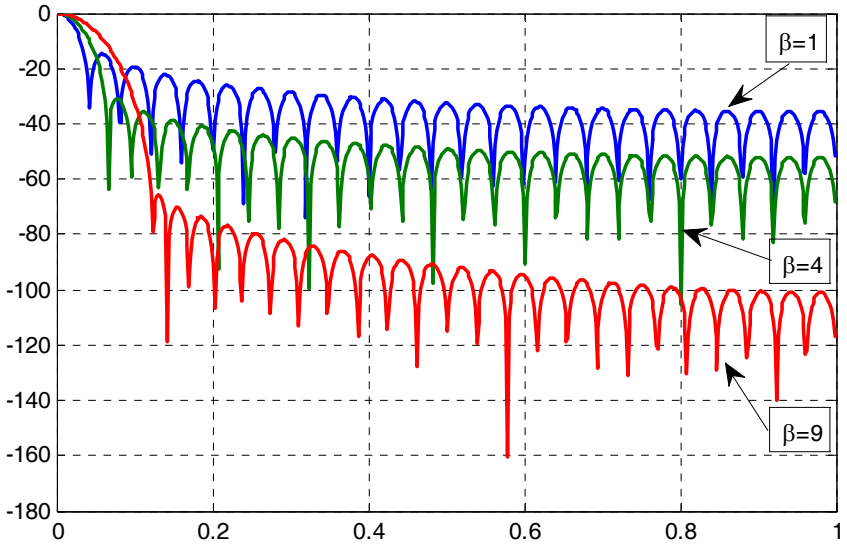


Figure E.25. Frequency domain plots for the Kaiser functions with L=50 and $\beta = 1$, $\beta = 4$, *and* $\beta = 9$

TABLE E.1 MATLAB Window Functions

Window Function	MATLAB Function
Bartlett	bartlett
Bartlett–Hann (modified)	barthannwin
Blackman	blackman
Blackman–Harris	blackmanharris
Bohman	bohmanwin
Chebyshev	chebwin
Flat Top	flattopwin
Gaussian	gausswin
Hamming	hamming
Hann or Hanning	hann
Kaiser	kaiser
Nuttall's Blackman–Harris	nuttallwin
Parzen (de la Valle–Poussin)	parzenwin
Rectangular	rectwin
Tukey	tukeywin
Triangular	triang

E.4 Fourier Series Method for Approximating an FIR Amplitude Response

The Fourier series method for approximating an FIR amplitude response is relatively straightforward, and is best used in conjunction with window functions. This is because the amplitude response $A(\omega)$ corresponding to a Discrete Time Linear Time Invariant (DTLTI) impulse response $h(n)$ is a periodic function of frequency and thus it can be expanded in a *Fourier series in the frequency domain.** The coefficients obtained from the Fourier series can then be related to the impulse response, and the desired coefficients of the FIR transfer function can then be obtained.

An advantage of FIR filter functions is the capability of obtaining linear phase (or constant time delay). To obtain time constant delay easily, it is necessary that the Fourier series in the frequency domain have either cosine terms only or sine terms only, but not both.

Let us consider the case of a cosine series representation and let the coefficients be expressed in normalized frequency $\nu = f/f_0$ where $f_0 = f_S/2$ is the folding frequency,† and f_S is the sampling frequency, i.e., $f_S = 1/T$. Also, let $A_d(\nu)$ be the desired amplitude response, and let $A(\nu)$ represent the approximation. Then, with this arrangement the period must always be selected as 2 units on the ν scale.

For the cosine series, the expression for $A(\nu)$ in exponential form is

$$A(\nu) = \sum_{m=-M}^{M} c_m e^{jm\pi\nu} \tag{E.14}$$

where M is a finite positive integer representing the terms after which the series is terminated, and c_m is the coefficient in exponential form that is computed from the integral

$$c_m = \int_0^1 A_d(\nu)\cos(m\pi\nu)d\nu \tag{E.15}$$

with $c_{-m} = c_m$

Relation (E.14) can be considered as the evaluation of some unknown discrete transfer function on the unit circle in the z–plane. We can perform this evaluation by letting $z = e^{j\omega T} = e^{jm\pi\nu}$.

Thus, if we denote this discrete transfer function as $H_1(z)$ and $e^{jm\pi\nu}$ is substituted for z, we obtain

* *In Chapter 7 we studied the application of Fourier series by expanding a periodic time function in the time domain.*

† *The folding frequency, also known as Nyquist frequency, is the highest frequency that can be represented in a digital signal of a specified sampling frequency. It is equal to oneóhalf of the sampling frequency.*

$$H_1(z) = \sum_{m=-M}^{M} c_m z^m \tag{E.16}$$

Relation (E.16) represents an FIR transfer function but it is non–causal because it includes positive powers of z, and this implies that the filter would produce an output advanced in time with respect to the input, and this is, of course, impossible. We can be overcome this problem by introducing a delay of M samples. Accordingly, we define the transfer function

$$H(z) = z^{-M}H_1(z) = z^{-M}\sum_{m=-M}^{M} c_m z^m = \sum_{m=M}^{-M} c_m z^{m-M} \tag{E.17}$$

In (E.17) we let $m - M = -i$. Then, with the substitution $c_{m-i} = a_i$ we obtain

$$H(z) = \sum_{i=0}^{2M} c_{m-i}\, z^{-i} = \sum_{i=0}^{2M} a_i z^{-i} \tag{E.18}$$

Let w_m denote the coefficients of a particular window function and let c'_m denote the coefficients of the modified function, i.e., the given function multiplied by some window function. The modified coefficients are then computed from

$$c'_m = w_m c_m \tag{E.19}$$

Example E.1

A low–pass FIR digital filter is to be designed using the Fourier series method. The desired amplitude response is

$$\begin{aligned} A(f) &= 1 \quad \text{for } 0 \le f \le 125 \text{ Hz} \\ &\quad 0 \quad \text{elsewhere in the range } 0 < f < f_0 \end{aligned} \tag{E.20}$$

The sampling frequency is 1 KHz and the impulse response is to be limited to 20 delays. Derive the transfer function using the Fourier series method.

Solution:

To make this low–pass filter an even function of frequency, we represent it as shown in Figure E.26 (a) in terms of the actual frequency. Let us express it in terms of the normalized frequency scale. We do this by considering the sampling frequency which is given as $f_S = 1$ KHz. Accordingly, the folding frequency* is $f_0 = f_S/2 = 1000/2 = 500$ Hz, and in terms of the normalized

* *We recall that the folding frequency is the highest frequency that can be processed by a given discrete–time system with sampling frequency* f_S

frequency, the folding frequency becomes 1, the sampling frequency becomes 2, and the cutoff frequency becomes $f_C = 125/500 = 0.25$. Therefore, (E.20) can be expressed as

$$\begin{aligned} A_d(\nu) &= 1 \quad \text{for } -0.25 < \nu < 0.25 \\ &0 \quad \text{elsewhere in the range } -1 < \nu < 1 \end{aligned} \tag{E.21}$$

and the low–pass filter in normalized frequency scale is as shown in Figure E.26(b).

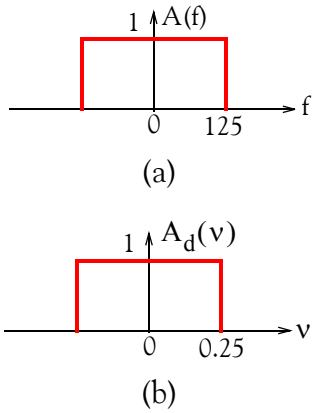


Figure E.26. Low–pass filter for Example E.1

The coefficients c_m are computed from the integral of (E.15),* that is,

$$c_m = \int_0^1 A_d(\nu)\cos(m\pi\nu)d\nu = \frac{\sin(m\pi\nu)}{m\pi}\Bigg|_0^{0.25} = \frac{\sin(0.25m\pi)}{m\pi} \tag{E.22}$$

with $c_{-m} = c_m$.

For this example, it is stated that the impulse response should be limited to 20 delays and this implies that the transfer function should contain 21 terms since one component needs not to be delayed. The coefficients c_m are computed from (E.22) for $m = 0$ through $m = 10$, and for the computations we use the MATLAB script below.

```
disp('m          cm'); disp('================'); m=0:10;
cm=zeros(11,2); % There are 11 terms from m=0 to m=10
cm(:,1)=m';
m=m+(m==0).*eps; % This statement avoids division by 0
cm(:,2)=sin(0.25.*m.*pi)./(m.*pi); fprintf('%2.0f\t %12.5f\n', cm')
```

MATLAB outputs the values shown below.

* *We recall from Chapter 7 that the Fourier series of even functions consist only of cosine terms.*

```
 m          cm
================
 0        0.25000
 1        0.22508
 2        0.15915
 3        0.07503
 4        0.00000
 5       -0.04502
 6       -0.05305
 7       -0.03215
 8       -0.00000
 9        0.02501
10        0.03183
```

Using the relation $c_{m-i} = a_i$,* for $0 \le i \le 2M$, we obtain the table below.

a_i	Rectangular window
a_0, a_{20}	0.03183
a_1, a_{19}	0.02501
a_2, a_{18}	0.00000
a_3, a_{17}	0.03215
a_4, a_{16}	0.05305
a_5, a_{15}	0.04502
a_6, a_{14}	0.00000
a_7, a_{13}	0.07503
a_8, a_{12}	0.15915
a_9, a_{11}	0.22508
a_{10}	0.25000

The window coefficients are applied to the non–causal form of the transfer function centered at the origin, and thus the 10th coefficient a_{10} is considered as the origin. These coefficients represent the unmodified transfer function and thus they correspond to the coefficients of the rectangular window.

* *Refer to equation (E.18)*

We will use the values shown below to plot the components of the transfer function H(z). These values are entered at the MATLAB command prompt.

```
a0=cm(11,2); a20=a0;
a1=cm(10,2); a19=a1;
a2=cm(9,2); a18=a2;
a3=cm(8,2); a17=a3;
a4=cm(7,2); a16=a4;
a5=cm(6,2); a15=a5;
a6=cm(5,2); a14=a6;
a7=cm(4,2); a13=a7;
a8=cm(3,2); a12=a8;
a9=cm(2,2); a11=a9;
a10=cm(1,2);
```

Then, the transfer function is expressed as

$$H(z) = \sum_{i=0}^{2M} a_i z^{-1} = \sum_{i=0}^{20} a_i z^{-1} = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{20} z^{-20} \tag{E.23}$$

Next, at the MATLAB command prompt we enter

```
AR=[a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20];
freqz(AR)
```

and we obtain the plot shown in Figure E.27.

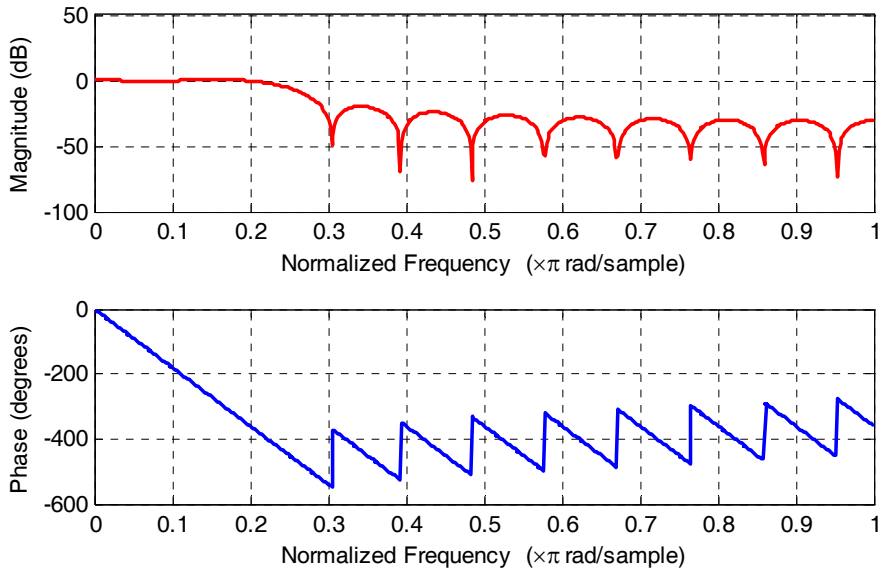


Figure E.27. The magnitude and phase response for the low–pass filter in Example E.1

Check with the fvtool MATLAB function:

```
b=0.25*sinc(0.25*(−10:10)); fvtool(b,1)
```

and MATLAB outputs the normalized magnitude response shown in Figure E.28.

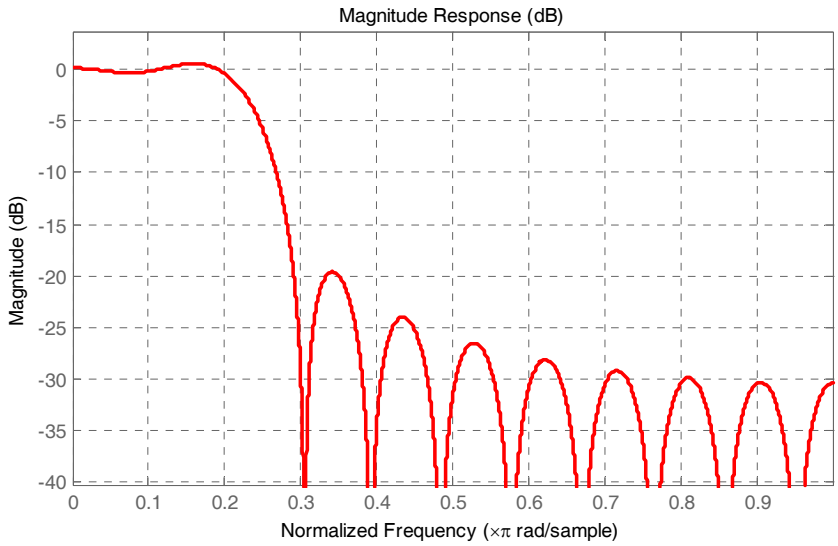


Figure E.28. Normalized frequency plot created with the MATLAB function fvtool

The MATLAB function fir1 uses the Fourier series approach in designing low–pass, high–pass, band–pass, and band elimination (band–stop) FIR digital filters. The syntax is

```
b=fir1(n,Wn,'ftype', window)
```

where n is the filter order, Wn is the cut–off frequency and must be between 0 < Wn < 1, with 1.0 corresponding to half the sample (Nyquist) rate, 'ftype' is the filter type specified as 'low', 'high', 'bandpass', or 'stop', and window is the window type, e.g., hann, blackman, and must be n+1 elements long. If no window is specified, the function fir1 uses a Hamming window.

The filter b is real and has linear phase. The normalized gain of the filter at Wn is –6 dB.

We can obtain the normalized frequency plot in Figure E.27 or E.28 using the fir1 function as follows:

```
b=fir1(20,0.25,'low', rectwin(21)); freqz(b)
```

and MATLAB outputs the magnitude and phase response shown in Figure E.29.*

* *The magnitude response is the same as that in Figure E.27 because the filter in this example is a "brick–wall" digital low–pass filter.*

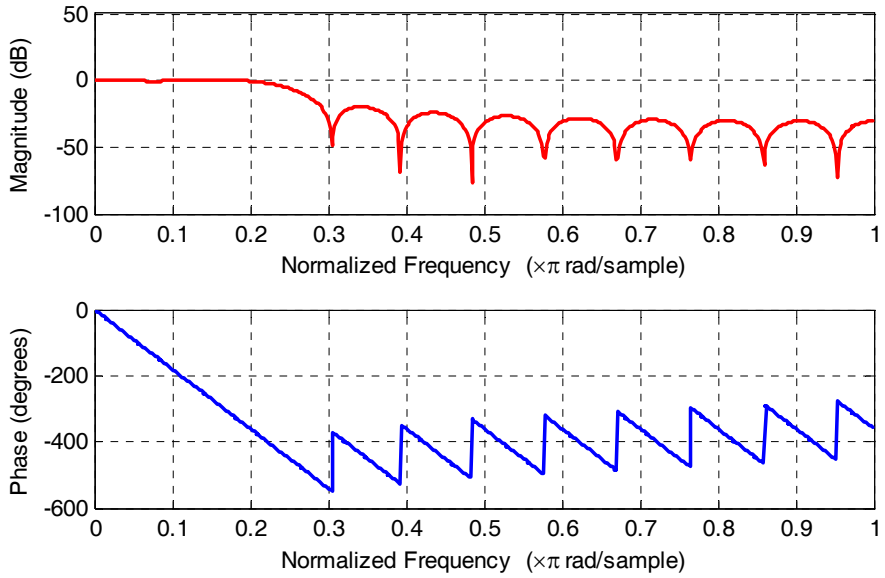


Figure E.29. Normalized frequency plots for the rectwin *window created with the MATLAB function* fir1

Example E.2

Compare the frequency response in Example E.1, Figure E.27, when it is multiplied by the following window functions:

a. triangular

b. Hanning

c. Hamming

Solution:

a. triangular

We recall from (E.4) that the triangular window function is defined as

$$f(t)_{triang} = 1 - \frac{2|t|}{\tau} \quad \text{for } |t| < \frac{\tau}{2}$$
$$= 0 \quad \text{otherwise} \tag{E.24}$$

Letting $t = mT$ and $\tau = 20T$ we obtain

$$W(m)_{triang} = 1 - \frac{2|mT|}{20T} = 1 - \frac{|m|}{10} \quad \text{for } |m| \le 10$$
$$= 0 \quad \text{otherwise} \tag{E.25}$$

As before, the window coefficients are applied to the non–causal form of the transfer function centered at the origin, and thus the 10th coefficient is considered as the origin. For the computations we use the MATLAB script below.

```
disp('m        wm')
disp('=================')
m=0:10; wm=zeros(11,2); wm(:,1)=m'; m=m+(m==0).*eps; wm(:,2)=1–m/10;
fprintf('%2.0f\t %12.5f\n',wm')
```

MATLAB outputs the values shown below.

```
m           wm
===============
0       1.00000
1       0.90000
2       0.80000
3       0.70000
4       0.60000
5       0.50000
6       0.40000
7       0.30000
8       0.20000
9       0.10000
10      0.00000
```

From (E.19)

$$c'_m = w_m c_m \tag{E.26}$$

For this example,

```
Cm=[0.2500 0.2251 0.1592 0.0750 0.0000 –0.0450 –0.0531 –0.0322 –0.0000 0.0250 0.0318];
Wm=[1.0000 0.9000 0.8000 0.7000 0.6000 0.5000 0.4000 0.3000 0.2000 0.1000 0.0000];
B=Cm.*Wm
```

and MATLAB outputs the modified coefficients below.

```
0.2500   0.2026   0.1274   0.0525   0   –0.0225   –0.0212   –0.0097   0   0.0025   0
```

and we add these coefficients in the table below.

a_i	Rectangular	Triangular
a_0, a_{20}	0.03183	0
a_1, a_{19}	0.02501	0.0025
a_2, a_{18}	0.00000	0
a_3, a_{17}	0.03215	–0.0097
a_4, a_{16}	0.05305	–0.0212
a_5, a_{15}	0.04502	–0.0225
a_6, a_{14}	0.00000	0
a_7, a_{13}	0.07503	0.0525
a_8, a_{12}	0.15915	0.1274
a_9, a_{11}	0.22508	0.2026
a_{10}	0.25000	0.2500

Next, we enter the a_i values at the MATLAB command prompt as shown below.

```
a0=wm(11,2)*cm(11,2); a20=a0;
a1=wm(10,2)*cm(10,2); a19=a1;
a2=wm(9,2)*cm(9,2); a18=a2;
a3=wm(8,2)*cm(8,2); a17=a3;
a4=wm(7,2)*cm(7,2); a16=a4;
a5=wm(6,2)*cm(6,2); a15=a5;
a6=wm(5,2)*cm(5,2); a14=a6;
a7=wm(4,2)*cm(4,2); a13=a7;
a8=wm(3,2)*cm(3,2); a12=a8;
a9=wm(2,2)*cm(2,2); a11=a9;
a10=wm(1,2)*cm(1,2);
```

and with

```
AT=[a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20];
freqz(AT)
```

we obtain the magnitude and phase response of the low–pass filter in Example E.1 modified with the triangular window function shown in Figure E.30.

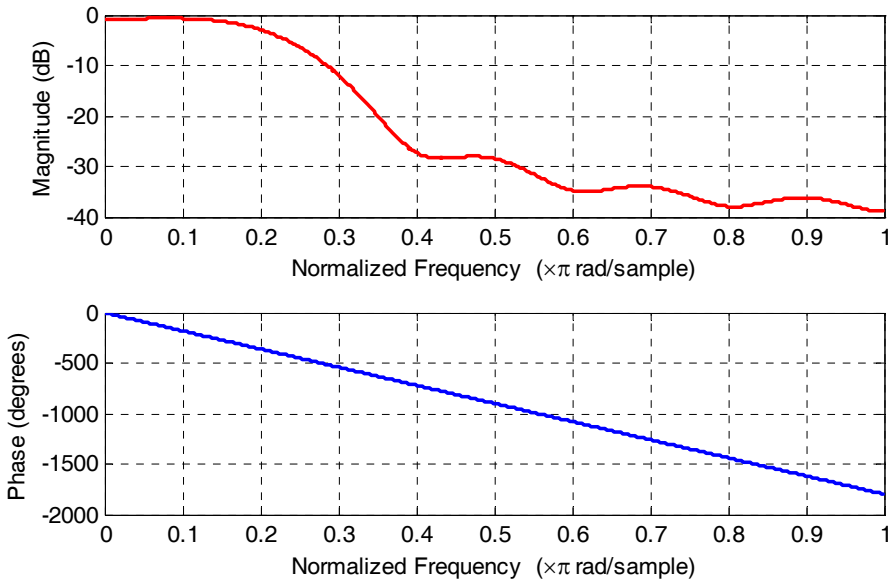


Figure E.30. Magnitude and phase response of the low–pass filter modified with the triangular window function

In comparison with the amplitude response with the rectangular window function in Figure E.29, the amplitude response with the triangular window function in Figure F.30 shows that the cutoff is worse than that in Figure E.29 but the sidelobes are reduced significantly.

We can obtain the normalized frequency plot in Figure E.30 using the **fir1** function as follows:

```
b=fir1(20,0.25,'low', triang(21)); freqz(b)
```

and MATLAB outputs the magnitude and phase response shown in Figure E.31.

Check with the **fvtool** MATLAB function:

```
b=0.25*sinc(0.25*(–10:10)); b=b.*triang(21)'; fvtool(b,1)
```

and MATLAB outputs the amplitude response shown in Figure E.32.

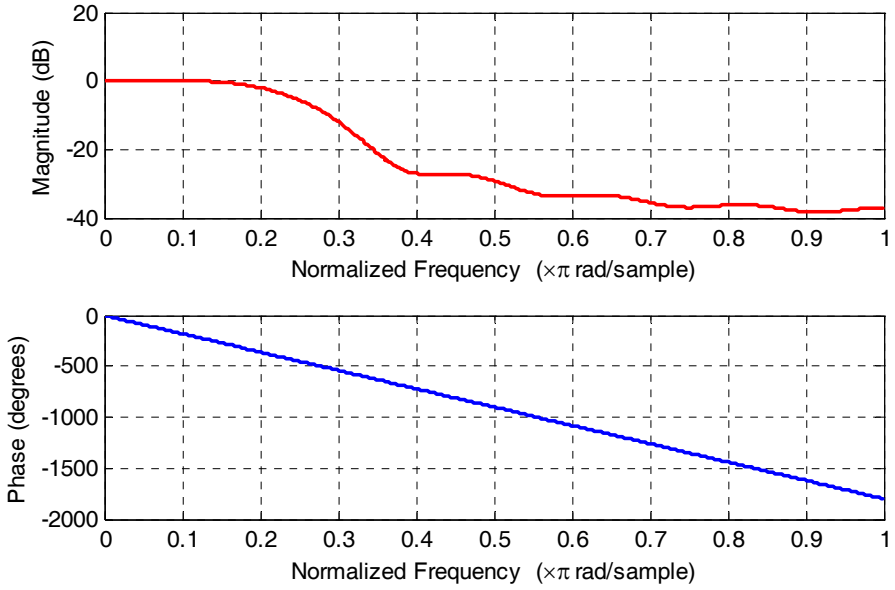


Figure E.31. Normalized frequency plots for the triang *window created with the MATLAB function* fir1

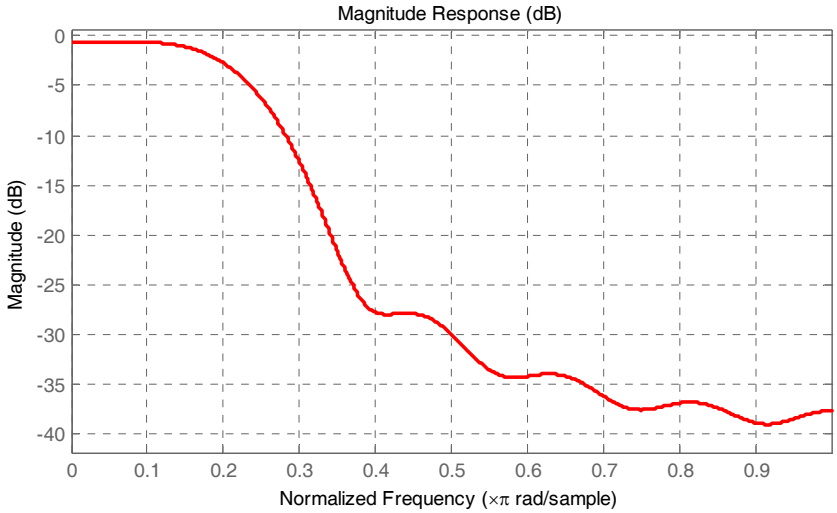


Figure E.32. Magnitude response for the triang *window created with the MATLAB function* fvtool

b. Hanning

We recall from (E.6) that the Hanning window function is defined as

$$f(t)_{Hann} = \cos^2(\pi t/\tau) = \frac{1}{2}\left(1 + \cos\frac{2\pi t}{\tau}\right) \quad \text{for } |t| < \frac{\tau}{2}$$
$$= 0 \quad \text{otherwise} \tag{E.27}$$

Letting $t = mT$ and $\tau = 20T$ we obtain

$$w(m)_{\text{Hann}} = \frac{1}{2}\left(1 + \cos\frac{2\pi|m|T}{20T}\right) = \frac{1}{2}\left(1 + \cos\frac{\pi|m|}{10}\right) \quad \text{for } |m| \le 10$$
$$= 0 \quad \text{otherwise} \tag{E.28}$$

As before, the window coefficients are applied to the non–causal form of the transfer function centered at the origin, and thus the 10th coefficient is considered as the origin. For the computations we use the MATLAB script below.

```
disp('m          wm')
disp('=================')
m=0:10; wm=zeros(11,2); wm(:,1)=m'; m=m+(m==0).*eps;
wm(:,2)=0.5.*(1 +cos(pi.*m./10));
fprintf('%2.0f\t %12.5f\n',wm')
```

and MATLAB outputs

```
m              wm
=================
0       1.00000
1       0.97553
2       0.90451
3       0.79389
4       0.65451
5       0.50000
6       0.34549
7       0.20611
8       0.09549
9       0.02447
10      0.00000
```

From (E.19)

$$c'_m = w_m c_m \tag{E.29}$$

Then,

```
Cm=[0.2500 0.2251 0.1592 0.0750 0.0000 –0.0450 –0.0531 –0.0322 –0.0000 0.0250
0.0318];
Wm=[1.00000 0.97553 0.90451 0.79389 0.65451 0.50000 0.34549 0.20611 0.09549
0.02447 0.00000];
```

Next,

```
C=Cm.*Wm
```

and MATLAB outputs

```
C =
0.2500 0.2196 0.1440 0.0595 0 -0.0225 -0.0183 -0.0066 0 0.0006 0
```

We add these coefficients in our table shown below.

a_i	Rectangular	Triangular	Hanning
a_0, a_{20}	0.03183	0	0
a_1, a_{19}	0.02501	0.0025	0.0006
a_2, a_{18}	0.00000	0	0
a_3, a_{17}	0.03215	–0.0097	–0.0066
a_4, a_{16}	0.05305	–0.0212	–0.0183
a_5, a_{15}	0.04502	–0.0225	–0.0225
a_6, a_{14}	0.00000	0	0
a_7, a_{13}	0.07503	0.0525	0.0595
a_8, a_{12}	0.15915	0.1274	0.1440
a_9, a_{11}	0.22508	0.2026	02196
a_{10}	0.25000	0.2500	0.2500

Next, we enter the a_i values at the MATLAB command prompt as shown below.

```
a0=wm(11,2)*cm(11,2); a20=a0;
a1=wm(10,2)*cm(10,2); a19=a1;
a2=wm(9,2)*cm(9,2); a18=a2;
a3=wm(8,2)*cm(8,2); a17=a3;
a4=wm(7,2)*cm(7,2); a16=a4;
a5=wm(6,2)*cm(6,2); a15=a5;
a6=wm(5,2)*cm(5,2); a14=a6;
a7=wm(4,2)*cm(4,2); a13=a7;
a8=wm(3,2)*cm(3,2); a12=a8;
a9=wm(2,2)*cm(2,2); a11=a9;
a10=wm(1,2)*cm(1,2);
```

and with

```
AHn=[a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20];
freqz(AHn)
```

we obtain the magnitude and phase response of the low–pass filter in Example E.1 modified with the Hanning window function shown in Figure E.33.

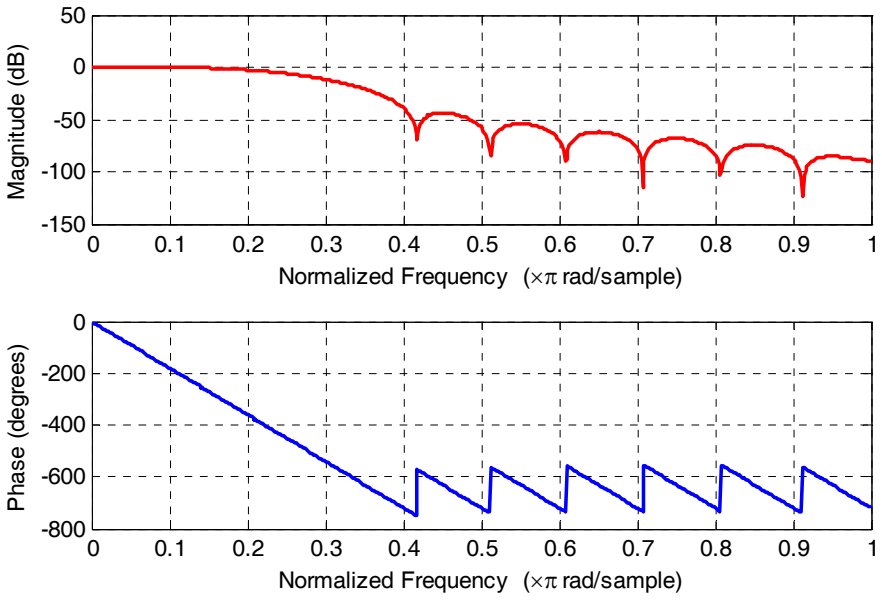


Figure E.33. Magnitude and phase response of the low–pass filter modified with the Hanning window function

We can obtain the normalized frequency plot in Figure E.30 using the **fir1** function as follows:

```
b=fir1(20,0.25,'low', hann(21)); freqz(b)
```

and MATLAB outputs the magnitude and phase response shown in Figure E.34.

Check with the **fvtool** MATLAB function:

```
b=0.25*sinc(0.25*(–10:10)); b=b.*hann(21)'; fvtool(b,1)
```

and MATLAB outputs the amplitude response shown in Figure E.35.

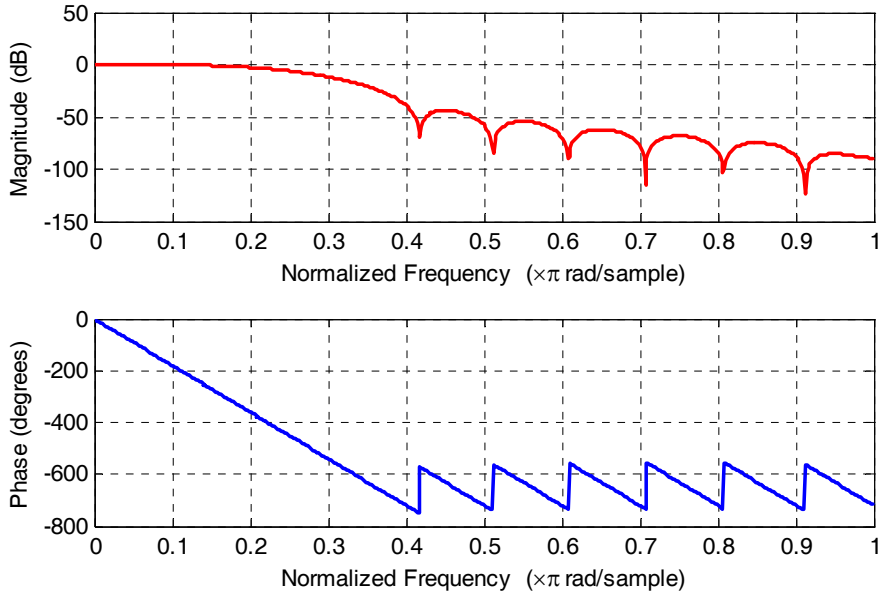


Figure E.34. Normalized frequency plots for the hann *window created with the MATLAB function* fir1

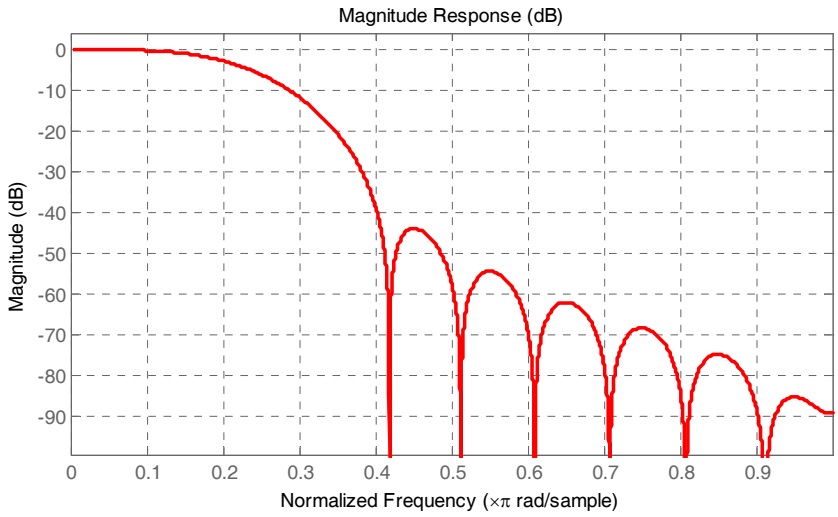


Figure E.35. Magnitude response for the hunn *window created with the MATLAB function* fvtool

c. Hamming

We recall from (E.8) that the Hamming window function is defined as

$$\begin{aligned} f(t)_{\text{Hamm}} &= 0.54 + 0.46\cos\frac{2\pi t}{\tau} \qquad \text{for } |t| < \frac{\tau}{2} \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{E.30}$$

Letting $t = mT$ and $\tau = 20T$ we obtain

$$w(m)_{Hamm} = 0.54 + 0.46\cos\frac{2\pi|m|T}{20T} = 0.54 + 0.46\cos\frac{\pi|m|}{10} \quad \text{for } |m| \le 10$$
$$= 0 \quad \text{otherwise} \tag{E.31}$$

As before, the window coefficients are applied to the non–causal form of the transfer function centered at the origin, and thus the 10th coefficient is considered as the origin. For the computations we use the MATLAB script below.

```
disp('m        wm')
disp('=================')
m=0:10; wm=zeros(11,2); wm(:,1)=m'; m=m+(m==0).*eps;
wm(:,2)=0.54+0.46.*(cos(pi.*m./10));
fprintf('%2.0f\t %12.5f\n',wm')
```

and MATLAB outputs

```
m          wm
==============
0       1.00000
1       0.97749
2       0.91215
3       0.81038
4       0.68215
5       0.54000
6       0.39785
7       0.26962
8       0.16785
9       0.10251
10      0.08000
```

From (E.19)

$$c'_m = w_m c_m \tag{E.32}$$

Then,

```
Cm=[0.2500 0.2251 0.1592 0.0750 0.0000 –0.0450 –0.0531 –0.0322 –0.0000 0.0250
0.0318];
Wm=[1.00000 0.97749 0.91215 0.81038 0.68215 0.54000 0.39785 0.26962 0.16785
0.10251 0.08000];
```

Next,

```
D=Cm.*Wm
```

and MATLAB outputs

```
D =
0.2500 0.2200 0.1452 0.0608 0 -0.0243 -0.0211 -0.0087 0  0.0026
0.0025
```

and we add these coefficients in our table shown below.

a_i	Rectangular	Triangular	Hanning	Hamming
a_0, a_{20}	0.03183	0	0	0.0025
a_1, a_{19}	0.02501	0.0025	0.0006	0.0026
a_2, a_{18}	0.00000	0	0	0
a_3, a_{17}	0.03215	–0.0097	–0.0066	–0.0087
a_4, a_{16}	0.05305	–0.0212	–0.0183	–0.0211
a_5, a_{15}	0.04502	–0.0225	–0.0225	–0.0243
a_6, a_{14}	0.00000	0	0	0
a_7, a_{13}	0.07503	0.0525	0.0595	0.0608
a_8, a_{12}	0.15915	0.1274	0.1440	0.1452
a_9, a_{11}	0.22508	0.2026	02196	0.2200
a_{10}	0.25000	0.2500	0.2500	0.2500

Next, we enter the a_i values at the MATLAB command prompt as shown below.

```
a0=wm(11,2)*cm(11,2); a20=a0;
a1=wm(10,2)*cm(10,2); a19=a1;
a2=wm(9,2)*cm(9,2); a18=a2;
a3=wm(8,2)*cm(8,2); a17=a3;
a4=wm(7,2)*cm(7,2); a16=a4;
a5=wm(6,2)*cm(6,2); a15=a5;
a6=wm(5,2)*cm(5,2); a14=a6;
a7=wm(4,2)*cm(4,2); a13=a7;
a8=wm(3,2)*cm(3,2); a12=a8;
a9=wm(2,2)*cm(2,2); a11=a9;
a10=wm(1,2)*cm(1,2);
```

and with

```
AHm=[a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20];
freqz(AHm)
```

we obtain the magnitude and phase response of the low–pass filter in Example E.1 modified with the Hanning window function shown in Figure E.36.

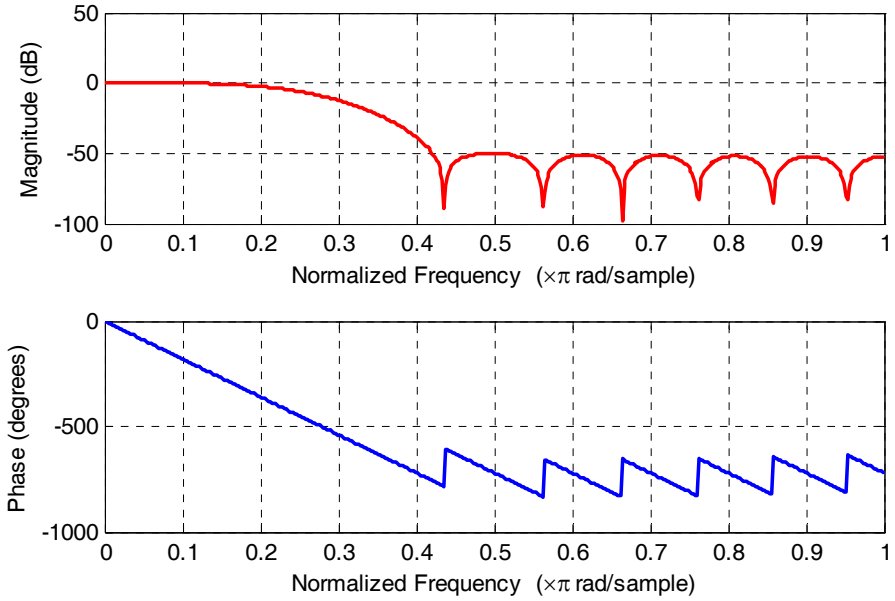


Figure E.36. Magnitude and phase response of the low–pass filter modified with the Hamming window function

We can obtain the normalized frequency plot in Figure E.36 using the **fir1** function* as follows:

```
b=fir1(20,0.25,'low'); freqz(b)
```

and MATLAB outputs the magnitude and phase response shown in Figure E.37.

Check with the **fvtool** MATLAB function:

```
b=0.25*sinc(0.25*(–10:10)); b=b.*hamming(21)'; fvtool(b,1)
```

and MATLAB outputs the amplitude response shown in Figure E.38.

* *As stated earlier, if we do not specify a window,* **fir1** *applies a Hamming window.*

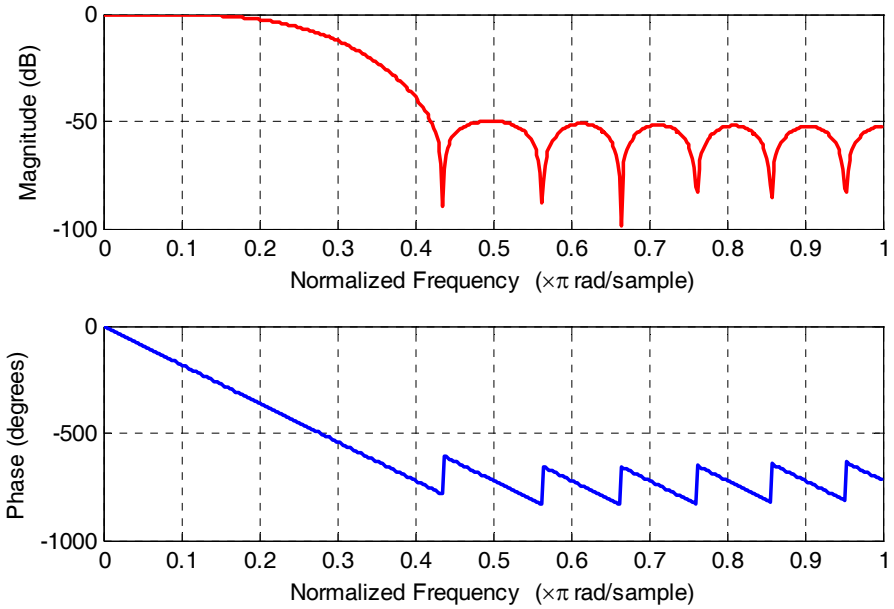


Figure E.37. Normalized frequency plots for the hamming *window created with the MATLAB function* fir1

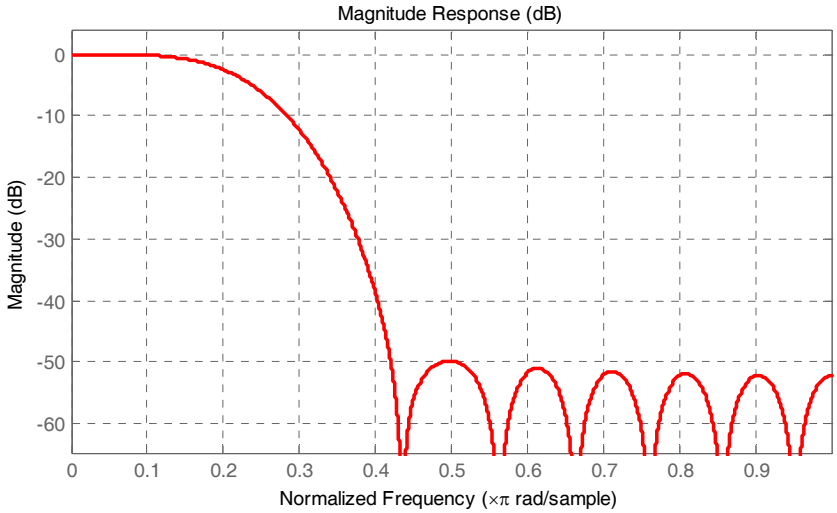


Figure E.38. Magnitude response for the hamming *window created with the MATLAB function* fvtool

Example E.3

Obtain the magnitude response for the low–pass filter in Example E.1 with Kaiser $\beta = 2\pi$ using:

a. the fir1 MATLAB function

b. the fvtool MATLAB function

Solution:

a.

```
b=fir1(20,0.25,'low', kaiser(21,2.*pi)); freqz(b)
```

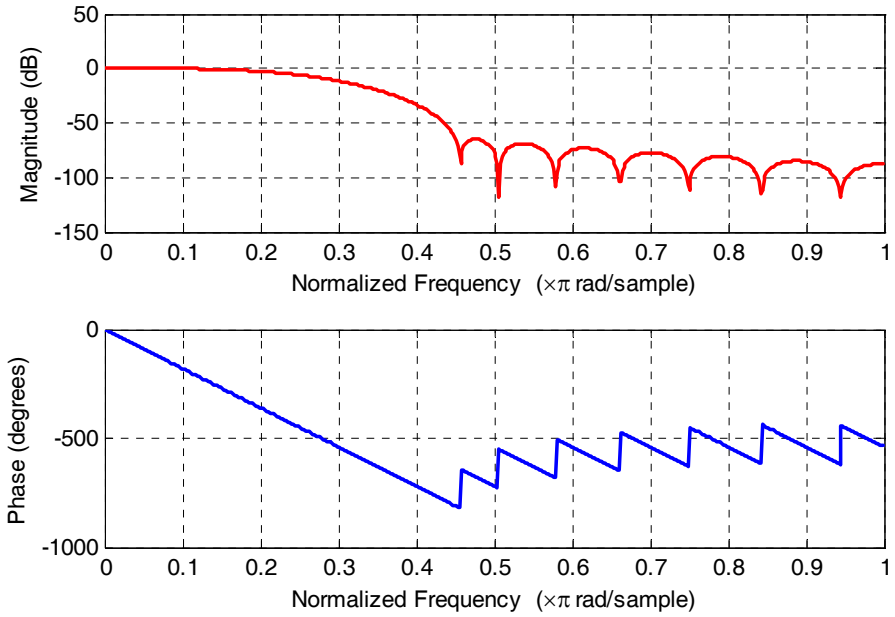


Figure E.39. Normalized frequency plots for the kaiser *window created with the MATLAB function* fir1

b.

```
b=0.25*sinc(0.25*(–10:10)); b=b.*kaiser(21,2.*pi)'; fvtool(b,1)
```

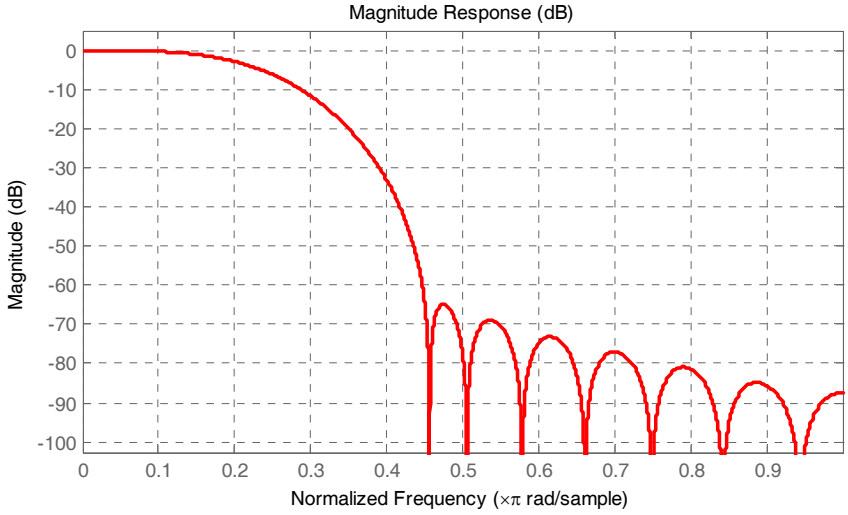


Figure E.40. Magnitude response for the kaiser *window created with the MATLAB function* fvtool

References and Suggestions for Further Study

A. The following publications by The MathWorks, are highly recommended for further study. They are available from The MathWorks, 3 Apple Hill Drive, Natick, MA, 01760, www.mathworks.com.

1. *Getting Started with MATLAB®*
2. *Using MATLAB®*
3. *Using MATLAB® Graphics*
4. *Using Simulink®*
5. *Sim Power Systems for Use with Simulink®*
6. *Fixed–Point Toolbox*
7. *Simulink® Fixed–Point*
8. *Real–Time Workshop*
9. *Signal Processing Toolbox*
10. *Getting Started with Signal Processing Blockset*
10. *Signal Processing Blockset*
11. *Control System Toolbox*
12. *Stateflow®*

B. Other references indicated in text pages and footnotes throughout this text, are listed below.

1. *Mathematics for Business, Science, and Technology,* ISBN 978–1–934404–01–0
2. *Numerical Analysis Using MATLAB® and Excel*, ISBN 978–1–934404–03–4
3. *Circuit Analysis I with MATLAB® Applications*, ISBN 978–0–9709511–2–0
4. *Circuit Analysis II with MATLAB® Applications*, ISBN 978–0–9709511–5–1
5. *Electronic Devices and Amplifier Circuits with MATLAB® Applications*, ISBN 978–0–9709511–7–5
6. *Digital Circuit Analysis and Design with Simulink Applications and Introduction to CPLDs and FPGAs*, ISBN 978–1–934404–05–8
7. *Introduction to Simulink® with Engineering Applications*, ISBN 978–1–934404–09–6

8. *Introduction to Stateflow® with Applications*, ISBN 978–1–934404–07–2

9. *Reference Data for Radio Engineers*, ISBN 0–672–21218–8, Howard W. Sams & Co.

10. *Electronic Engineers' Handbook*, ISBN 0–07–020981–2, McGraw–Hill

11. *Network Analysis and Synthesis*, L. Weinberg, McGraw–Hill

12. *Elecrronic Filter Design Handbook*, Williams and Taylor, McGraw–Hill

Index