

Signals and Systems

Collection Editor:

Richard Baraniuk

Signals and Systems

Collection Editor:

Richard Baraniuk

Authors:

Thanos Antoulas
Richard Baraniuk
Steven Cox
Benjamin Fite
Roy Ha
Michael Haag
Matthew Hutchinson

Don Johnson
Ricardo Radaelli-Sanchez
Justin Romberg
Phil Schniter
Melissa Selik
JP Slavinsky

Online:

<<http://cnx.org/content/col10064/1.11/>>

C O N N E X I O N S

Rice University, Houston, Texas

©2008 Richard Baraniuk

This selection and arrangement of content is licensed under the Creative Commons Attribution License:
<http://creativecommons.org/licenses/by/1.0>

Table of Contents

1 Signals	
1.1 Signal Classifications and Properties	1
1.2 Size of A Signal: Norms	9
1.3 Signal Operations	19
1.4 Useful Signals	22
1.5 The Impulse Function	25
1.6 The Complex Exponential	28
1.7 Discrete-Time Signals	31
Solutions	35
2 Systems	
2.1 System Classifications and Properties	37
2.2 Properties of Systems	41
3 Time Domain Analysis of Continuous Time Systems	
3.1 CT Linear Systems and Differential Equations	47
3.2 Continuous-Time Convolution	53
3.3 Properties of Convolution	59
3.4 BIBO Stability	65
4 Time Domain Analysis of Discrete Time Systems	
4.1 Discrete-Time Systems in the Time-Domain	69
4.2 Discrete-Time Convolution	73
4.3 Circular Convolution and the DFT	158
4.4 Linear Constant-Coefficient Difference Equations	82
4.5 Solving Linear Constant-Coefficient Difference Equations	83
Solutions	89
5 Linear Algebra Overview	
5.1 Linear Algebra: The Basics	91
5.2 Eigenvectors and Eigenvalues	96
5.3 Matrix Diagonalization	101
5.4 Eigen-stuff in a Nutshell	104
5.5 Eigenfunctions of LTI Systems	105
5.6 Fourier Transform Properties	108
Solutions	109
6 Continuous Time Fourier Series	
6.1 Periodic Signals	111
6.2 Fourier Series: Eigenfunction Approach	112
6.3 Derivation of Fourier Coefficients Equation	115
6.4 Fourier Series in a Nutshell	116
6.5 Fourier Series Properties	119
6.6 Symmetry Properties of the Fourier Series	122
6.7 Circular Convolution Property of Fourier Series	126
6.8 Fourier Series and LTI Systems	127
6.9 Convergence of Fourier Series	130
6.10 Dirichlet Conditions	132
6.11 Gibbs's Phenomena	134
6.12 Fourier Series Wrap-Up	137

Solutions	139
7 Discrete Fourier Transform	
7.1 Fourier Analysis	141
7.2 Fourier Analysis in Complex Spaces	142
7.3 Matrix Equation for the DTFS	149
7.4 Periodic Extension to DTFS	150
7.5 Circular Shifts	154
7.6 Circular Convolution and the DFT	158
Solutions	163
8 Fast Fourier Transform (FFT)	
8.1 DFT: Fast Fourier Transform	165
8.2 The Fast Fourier Transform (FFT)	166
8.3 Deriving the Fast Fourier Transform	167
Solutions	170
9 Convergence	
9.1 Convergence of Sequences	171
9.2 Convergence of Vectors	173
9.3 Uniform Convergence of Function Sequences	176
10 Discrete Time Fourier Transform (DTFT)	
10.1 Discrete Fourier Transformation	179
10.2 Discrete Fourier Transform (DFT)	181
10.3 Table of Common Fourier Transforms	183
10.4 Discrete-Time Fourier Transform (DTFT)	184
10.5 Discrete-Time Fourier Transform Properties	185
10.6 Discrete-Time Fourier Transform Pair	185
10.7 DTFT Examples	186
Solutions	190
11 Continuous Time Fourier Transform (CTFT)	
11.1 Continuous-Time Fourier Transform (CTFT)	191
11.2 Properties of the Continuous-Time Fourier Transform	192
Solutions	196
12 Sampling Theorem	
12.1 Sampling	197
12.2 Reconstruction	201
12.3 More on Reconstruction	205
12.4 Nyquist Theorem	207
12.5 Aliasing	209
12.6 Anti-Aliasing Filters	212
12.7 Discrete Time Processing of Continuous Time Signals	214
Solutions	217
13 Laplace Transform and System Design	
13.1 The Laplace Transforms	219
13.2 Properties of the Laplace Transform	222
13.3 Table of Common Laplace Transforms	223
13.4 Region of Convergence for the Laplace Transform	223
13.5 The Inverse Laplace Transform	225
13.6 Poles and Zeros	227

14 Z-Transform and Digital Filtering	
14.1 The Z Transform: Definition	231
14.2 Table of Common z-Transforms	236
14.3 Region of Convergence for the Z-transform	237
14.4 Inverse Z-Transform	246
14.5 Rational Functions	249
14.6 Difference Equation	251
14.7 Understanding Pole/Zero Plots on the Z-Plane	254
14.8 Filter Design using the Pole/Zero Plot of a Z-Transform	258
15 Appendix: Hilbert Spaces and Orthogonal Expansions	
15.1 Vector Spaces	263
15.2 Norms	265
15.3 Inner Products	268
15.4 Hilbert Spaces	270
15.5 Cauchy-Schwarz Inequality	270
15.6 Common Hilbert Spaces	277
15.7 Types of Basis	280
15.8 Orthonormal Basis Expansions	283
15.9 Function Space	287
15.10 Haar Wavelet Basis	288
15.11 Orthonormal Bases in Real and Complex Spaces	295
15.12 Plancharel and Parseval's Theorems	297
15.13 Approximation and Projections in Hilbert Space	298
Solutions	301
16 Homework Sets	
16.1 Homework 1	303
16.2 Homework 1 Solutions	307
17 Viewing Embedded LabVIEW Content	319
Glossary	320
Index	325
Attributions	330

Chapter 1

Signals

1.1 Signal Classifications and Properties¹

1.1.1 Introduction

This module will lay out some of the fundamentals of signal classification. This is basically a list of definitions and properties that are fundamental to the discussion of signals and systems. It should be noted that some discussions like energy signals vs. power signals² have been designated their own module for a more complete discussion, and will not be included here.

1.1.2 Classifications of Signals

Along with the classification of signals below, it is also important to understand the Classification of Systems (Section 2.1).

1.1.2.1 Continuous-Time vs. Discrete-Time

As the names suggest, this classification is determined by whether or not the time axis (x-axis) is **discrete** (countable) or **continuous** (Figure 1.1). A continuous-time signal will contain a value for all real numbers along the time axis. In contrast to this, a discrete-time signal (Section 1.7) is often created by using the sampling theorem³ to sample a continuous signal, so it will only have values at equally spaced intervals along the time axis.

¹This content is available online at <http://cnx.org/content/m10057/2.17/>.

²"Signal Energy vs. Signal Power" <http://cnx.org/content/m10055/latest/>

³"The Sampling Theorem" <http://cnx.org/content/m0050/latest/>

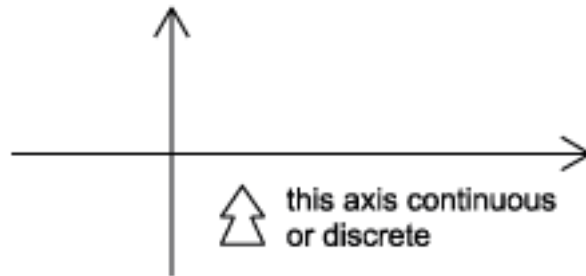


Figure 1.1

1.1.2.2 Analog vs. Digital

The difference between **analog** and **digital** is similar to the difference between continuous-time and discrete-time. In this case, however, the difference is with respect to the value of the function (y-axis) (Figure 1.2). Analog corresponds to a continuous y-axis, while digital corresponds to a discrete y-axis. An easy example of a digital signal is a binary sequence, where the values of the function can only be one or zero.

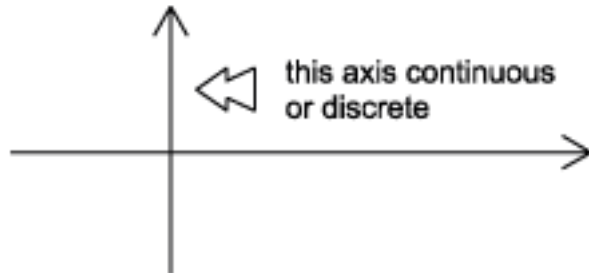


Figure 1.2

1.1.2.3 Periodic vs. Aperiodic

Periodic signals (Section 6.1) repeat with some **period** T , while aperiodic, or nonperiodic, signals do not (Figure 1.3). We can define a periodic function through the following mathematical expression, where t can be any number and T is a positive constant:

$$f(t) = f(T + t) \quad (1.1)$$

The **fundamental period** of our function, $f(t)$, is the smallest value of T that the still allows (1.1) to be true.

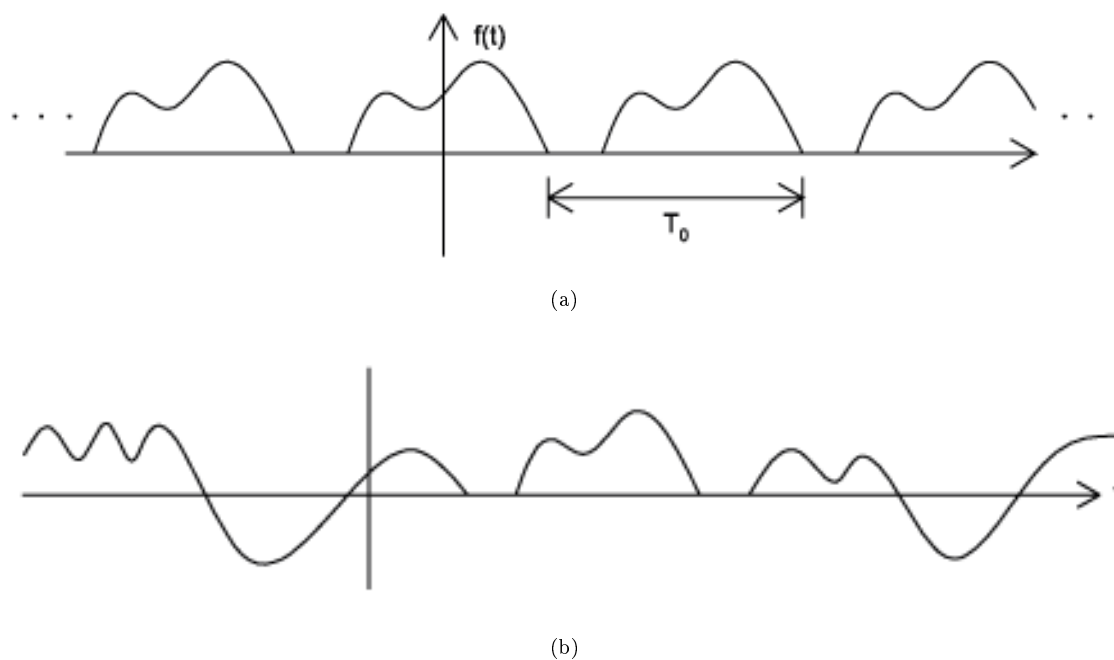
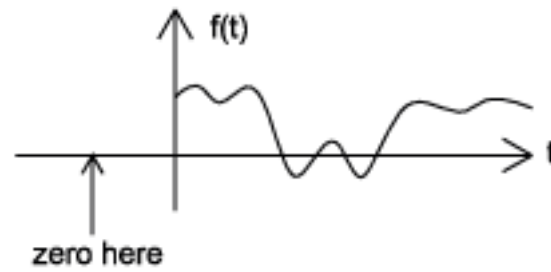


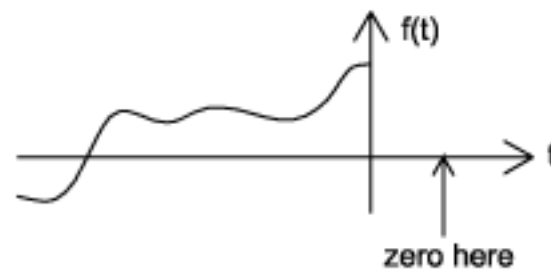
Figure 1.3: (a) A periodic signal with period T_0 (b) An aperiodic signal

1.1.2.4 Causal vs. Anticausal vs. Noncausal

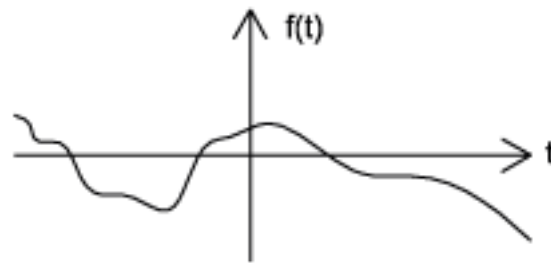
Causal signals are signals that are zero for all negative time, while **anticausal** are signals that are zero for all positive time. **Noncausal** signals are signals that have nonzero values in both positive and negative time (Figure 1.4).



(a)



(b)



(c)

Figure 1.4: (a) A causal signal (b) An anticausal signal (c) A noncausal signal

1.1.2.5 Even vs. Odd

An **even signal** is any signal f such that $f(t) = f(-t)$. Even signals can be easily spotted as they are *symmetric* around the vertical axis. An **odd signal**, on the other hand, is a signal f such that $f(t) = -f(-t)$ (Figure 1.5).

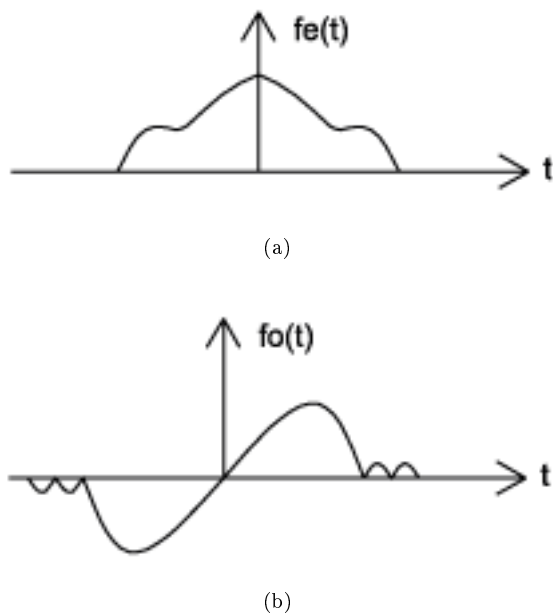


Figure 1.5: (a) An even signal (b) An odd signal

Using the definitions of even and odd signals, we can show that any signal can be written as a combination of an even and odd signal. That is, every signal has an odd-even decomposition. To demonstrate this, we have to look no further than a single equation.

$$f(t) = \frac{1}{2}(f(t) + f(-t)) + \frac{1}{2}(f(t) - f(-t)) \quad (1.2)$$

By multiplying and adding this expression out, it can be shown to be true. Also, it can be shown that $f(t) + f(-t)$ fulfills the requirement of an even function, while $f(t) - f(-t)$ fulfills the requirement of an odd function (Figure 1.6).

Example 1.1

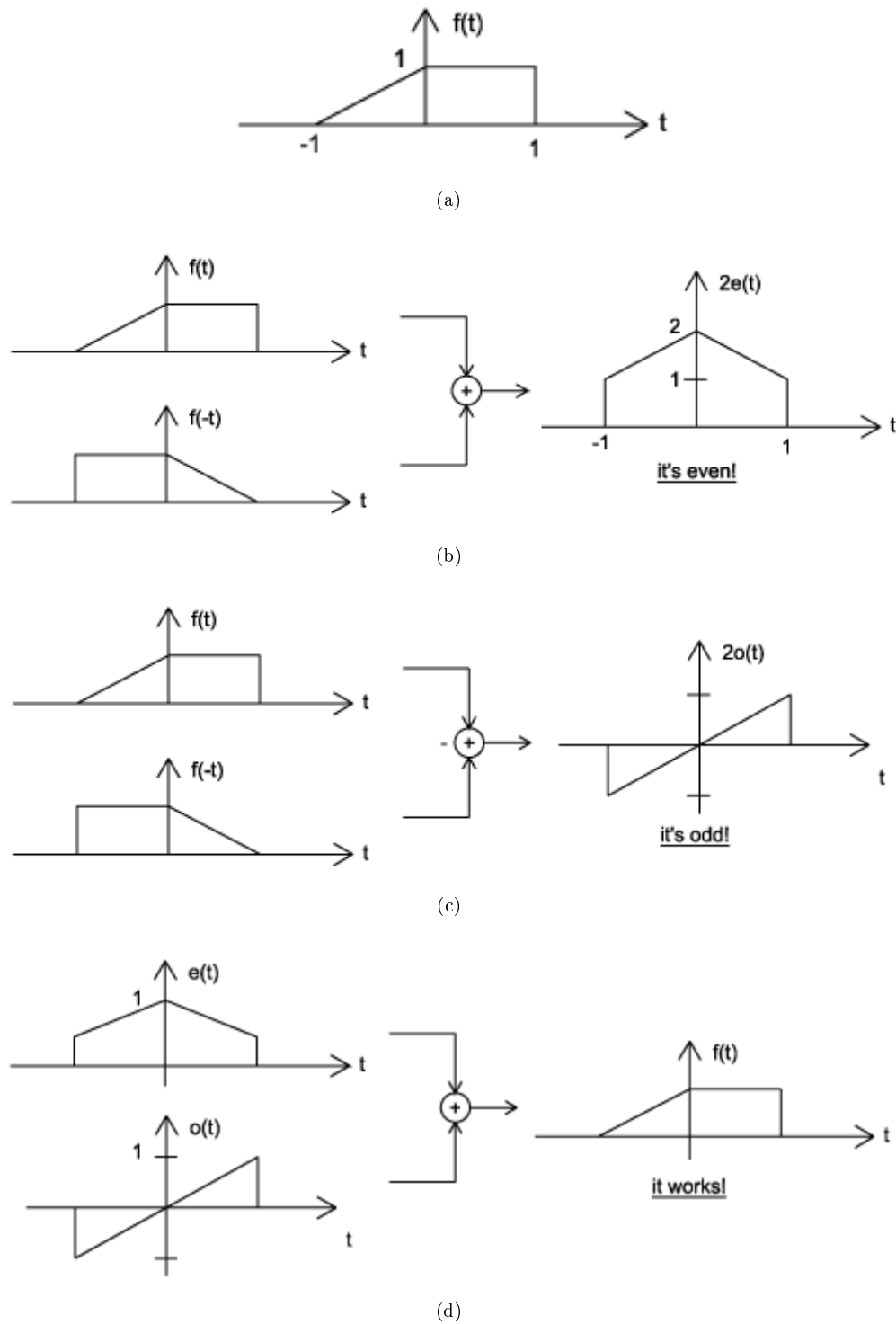


Figure 1.6: (a) The signal we will decompose using odd-even decomposition (b) Even part: $e(t) = \frac{1}{2}(f(t) + f(-t))$ (c) Odd part: $o(t) = \frac{1}{2}(f(t) - f(-t))$ (d) Check: $e(t) + o(t) = f(t)$

1.1.2.6 Deterministic vs. Random

A **deterministic signal** is a signal in which each value of the signal is fixed and can be determined by a mathematical expression, rule, or table. Because of this the future values of the signal can be calculated from past values with complete confidence. On the other hand, a **random signal**⁴ has a lot of uncertainty about its behavior. The future values of a random signal cannot be accurately predicted and can usually only be guessed based on the averages⁵ of sets of signals (Figure 1.7).

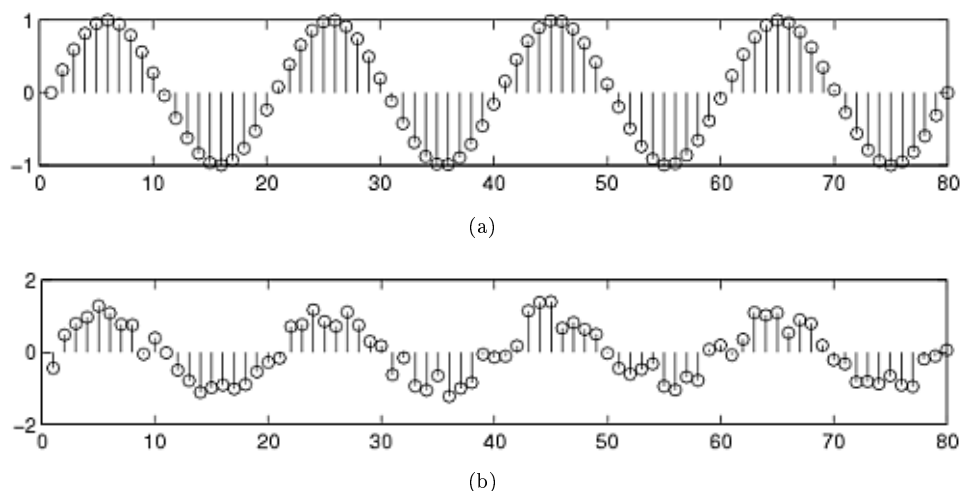


Figure 1.7: (a) Deterministic Signal (b) Random Signal

1.1.2.7 Right-Handed vs. Left-Handed

A **right-handed** signal and **left-handed** signal are those signals whose value is zero between a given variable and positive or negative infinity. Mathematically speaking, a right-handed signal is defined as any signal where $f(t) = 0$ for $t < t_1 < \infty$, and a left-handed signal is defined as any signal where $f(t) = 0$ for $t > t_1 > -\infty$. See (Figure 1.8) for an example. Both figures "begin" at t_1 and then extends to positive or negative infinity with mainly nonzero values.

⁴"Introduction to Random Signals and Processes" <<http://cnx.org/content/m10649/latest/>>

⁵"Random Processes: Mean and Variance" <<http://cnx.org/content/m10656/latest/>>

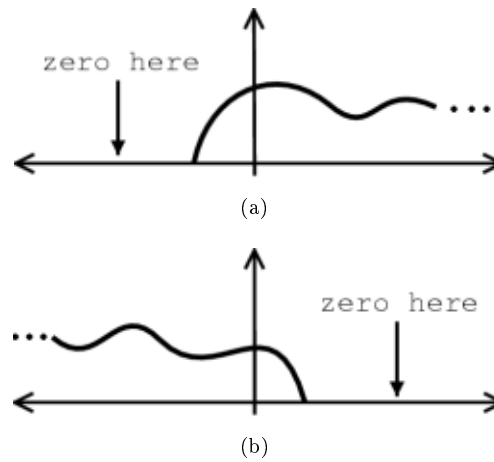


Figure 1.8: (a) Right-handed signal (b) Left-handed signal

1.1.2.8 Finite vs. Infinite Length

As the name applies, signals can be characterized as to whether they have a finite or infinite length set of values. Most finite length signals are used when dealing with discrete-time signals or a given sequence of values. Mathematically speaking, $f(t)$ is a **finite-length signal** if it is *nonzero* over a finite interval

$$t_1 < f(t) < t_2$$

where $t_1 > -\infty$ and $t_2 < \infty$. An example can be seen in Figure 1.9. Similarly, an **infinite-length signal**, $f(t)$, is defined as nonzero over all real numbers:

$$\infty \leq f(t) \leq -\infty$$

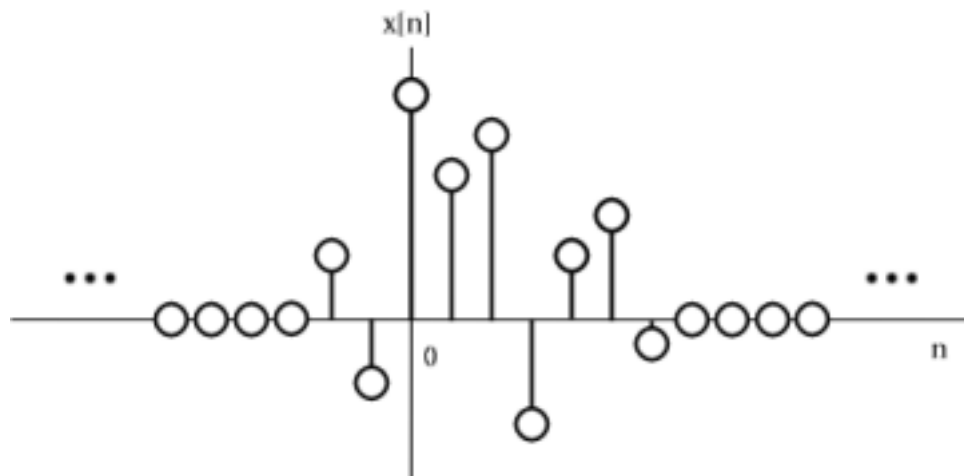


Figure 1.9: Finite-Length Signal. Note that it only has nonzero values on a set, finite interval.

1.2 Size of A Signal: Norms⁶

"Size" indicates *largeness* or *strength*. We will use the mathematical concept of the *norm* to quantify this notion for both continuous-time and discrete-time signals. First we consider a way to quantify the size of a signal which may already be familiar.

1.2.1 Continuous-Time Energy

Our usual notion of the energy of a signal is the area under the curve $(|f(t)|)^2$

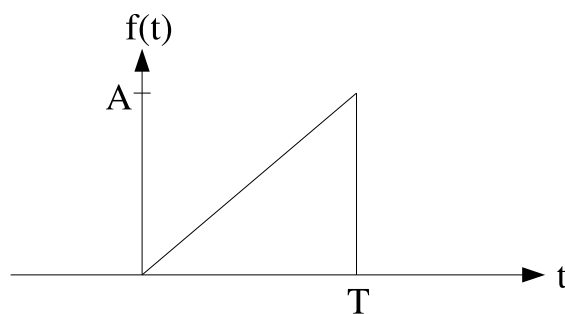


Figure 1.10

$$E_f = \int_{-\infty}^{\infty} (|f(t)|)^2 dt \quad (1.3)$$

⁶This content is available online at <<http://cnx.org/content/m12363/1.2/>>.

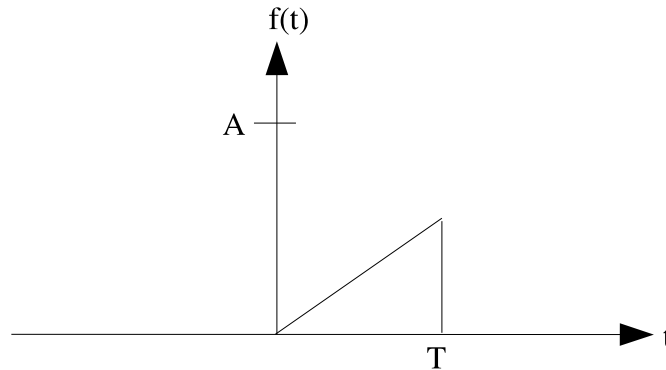
Example 1.2Calculate E_f for

Figure 1.11

1.2.2 Generalized Energy: NormsThe notion of "energy" can be generalized through the introduction of the L_p norm:

$$\|f\|_p = \left(\int (|f(t)|)^p dt \right)^{\frac{1}{p}} \quad (1.4)$$

where $1 \leq p < \infty$.**Example 1.3**

$$E_f = (\|f\|_2)^2$$

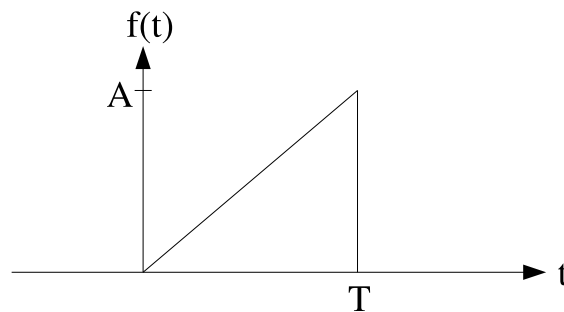
Example 1.4Calculate the L_p norm of

Figure 1.12

Exercise 1.1

What happens to

$$\|f\|_p = \left(\int (|f(t)|)^p dt \right)^{\frac{1}{p}}$$

as $p \rightarrow \infty$?

L_∞ norm

$$\|f\|_\infty = \text{ess sup}|f(t)|$$

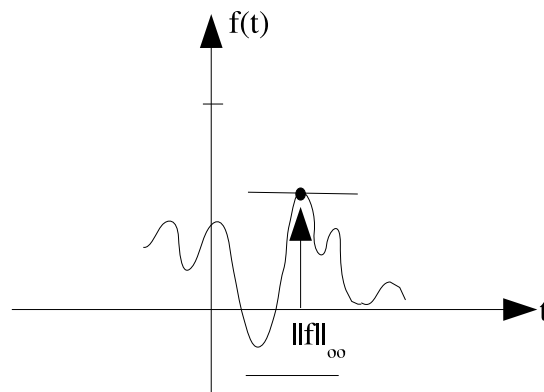


Figure 1.13

1.2.3 Discrete-Time Energy

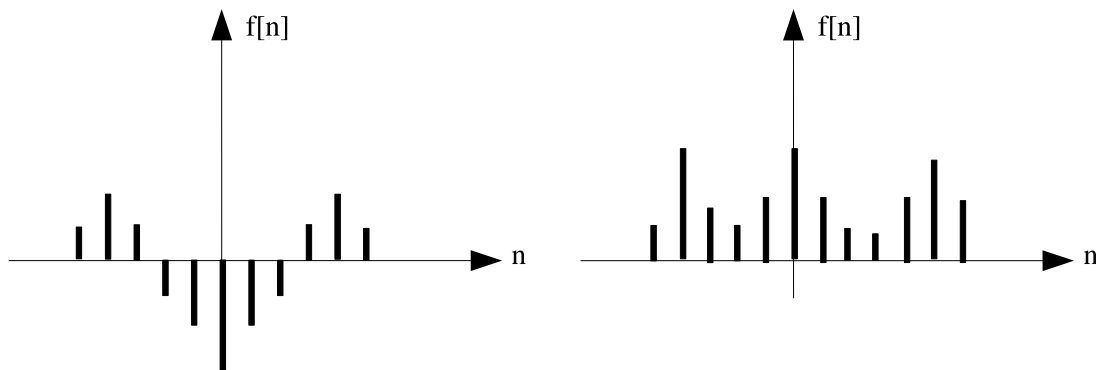


Figure 1.14

$$E_f = \sum_{n=-\infty}^{\infty} (|f[n]|)^2$$

$$\|f\|_p = \left(\int (|f(t)|)^p dt \right)^{\frac{1}{p}}$$

where $1 \leq p < \infty$

$$\|f\|_\infty = \max_n \{|f[n]|\}$$

$$\left(\|f\|_p\right)^p = \sum_{n=0}^{N-1} (|f[n]|)^p$$

where $1 \leq p < \infty$

$$\|f\|_\infty = \max_{n=0, 1, \dots, N-1} \{|f[n]|\}$$

1.2.4 Finite Norm Signals

What are the conditions on a signal for $\|f\|_p < \infty$? Look at all 4 fundamental signal classes

1.2.4.1 Discrete-Time and Finite Length

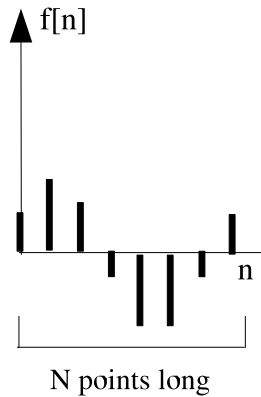


Figure 1.15

This is a length N vector.

$$\mathbf{f} = \begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ f[\dots] \\ f[N-1] \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \dots \\ f_{N-1} \end{pmatrix}$$

where $f \in \mathbb{C}^N$, or $f \in \mathbb{R}^N$ N -dimensional complex or real Euclidean space.

Example 1.5

$N = 3$, f is a real signal.

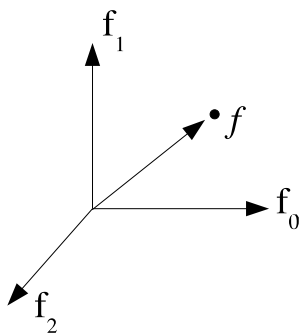


Figure 1.16

Definition 1:

$$l_p[0, N-1] = \{f \in \mathbb{C}^N, \|f\|_p < \infty\}$$

but from previous discussion $l_p[0, N-1] = \mathbb{C}^N$

1.2.4.2 Discrete-Time and Infinite Length

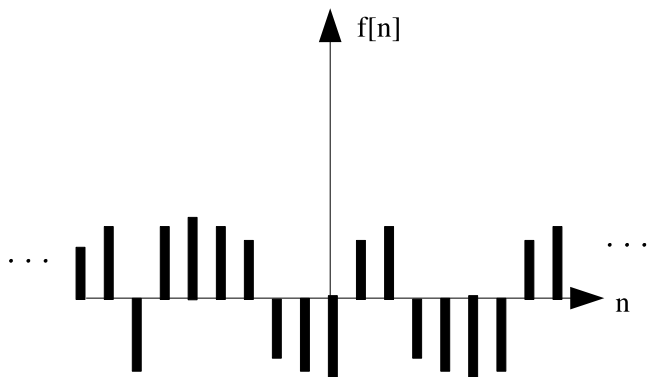


Figure 1.17

can still interpret f as an *infinite-length* vector

$$\mathbf{f} = \begin{pmatrix} f[\dots] \\ f[-1] \\ f[0] \\ f[1] \\ f[2] \\ f[\dots] \end{pmatrix}$$

but \mathbb{C}^∞ , R^∞ don't make sense.

Definition 2:

$$l_p(z) = \{f, \|f\|_p < \infty\}$$

$$\left(\|f\|_p\right)^p = \sum_{n=-\infty}^{\infty} (|f[n]|)^p$$

where $1 \leq p < \infty$

$$\|f\|_\infty = \max_{n \in z} \{|f[n]|\}$$

What does it take for an f to be in $l_p(z)$?

Example 1.6

Sketch an $f \in l_p(z)$ and $f \notin l_p(z)$.

Exercise 1.2

What characteristics does $f \in l_p(z)$ have and what happens as we change p ?

1.2.4.3 Continuous-Time and Finite-Length

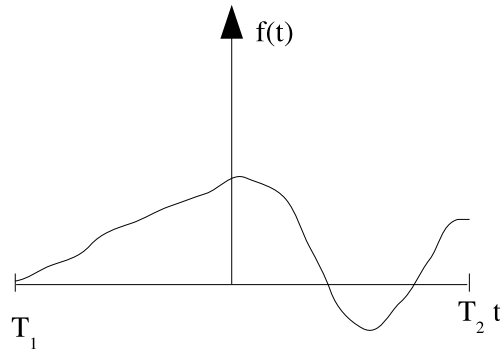


Figure 1.18

We will still refer to $f(t)$ as a vector; more on this later.

Definition 3:

$$L_p[T_1, T_2] = \{f[T_1, T_2], \|f\|_p < \infty\}$$

$$\|f\|_p = \left(\int_{T_1}^{T_2} (|f(t)|)^p dt \right)^{\frac{1}{p}}$$

where $1 \leq p < \infty$

$$\|f\|_\infty = \text{esssup}|f(t)|$$

where $T_1 \leq t \leq T_2$

Exercise 1.3

What does it take for and f to be in $L_p[T_1, T_2]$?

1.2.4.4 Continuous-Time and Infinite-Length

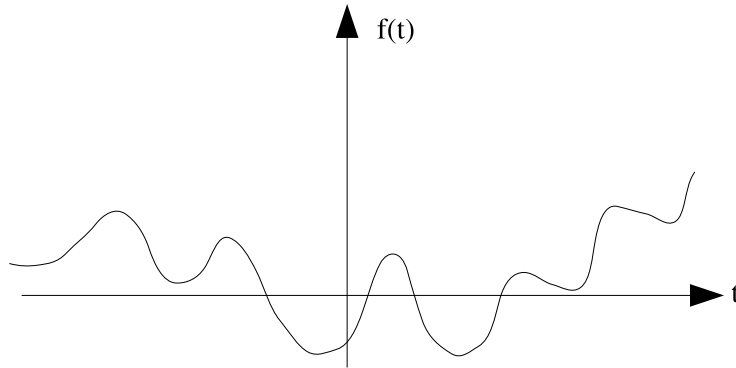


Figure 1.19

We will still refer to $f(t)$ as a *vector*.

Definition 4:

$$L_p(R) = \{f, \|f\|_p < \infty\}$$

$$\|f\|_p = \left(\int_{-\infty}^{\infty} (|f[n]|)^p dt \right)^{\frac{1}{p}}$$

where $1 \leq p < \infty$

$$\|f\|_{\infty} = \text{esssup}|f(t)|$$

where $-\infty < t < \infty$

Exercise 1.4

What does it take for an $f \in L_p(R)$?

Example 1.7

Sketch an $f \in L_p(R)$ and $f \notin L_p(R)$.

1.2.5 Power

What do we do when $\|f\|_p = \infty$?

Example 1.8: Periodic Signal

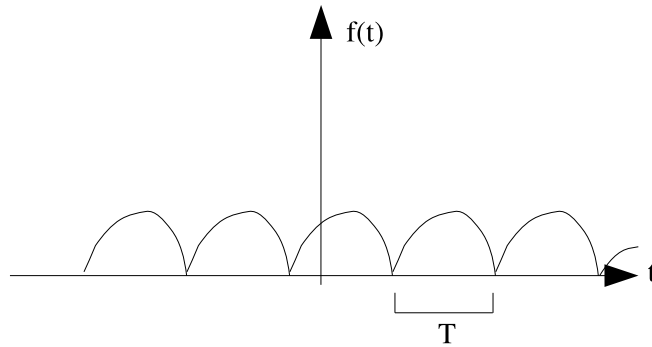


Figure 1.20

Solution: Look at the "norm per unit time".

- \mathbf{ie} - norm over one period.
- \mathbf{ie} - norm of infinite-length signal converted to finite length signal by windowing.

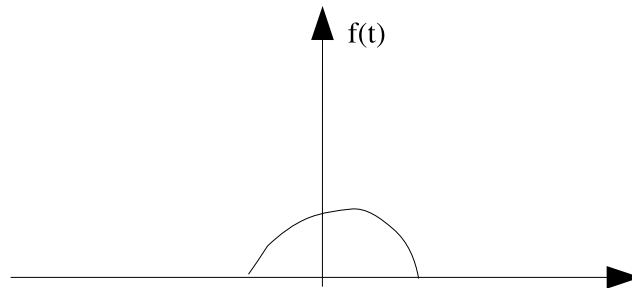


Figure 1.21

$\frac{\|f\|_p}{T}$ is the measure.

Units for $p = 2$?

L_2 Power = "energy per unit time"

- Useful when $E_f = \infty$
- time average of energy

$$P_f = \lim_{T \rightarrow \infty} \int_{-(\frac{T}{2})}^{\frac{T}{2}} (|f(t)|)^2 dt$$

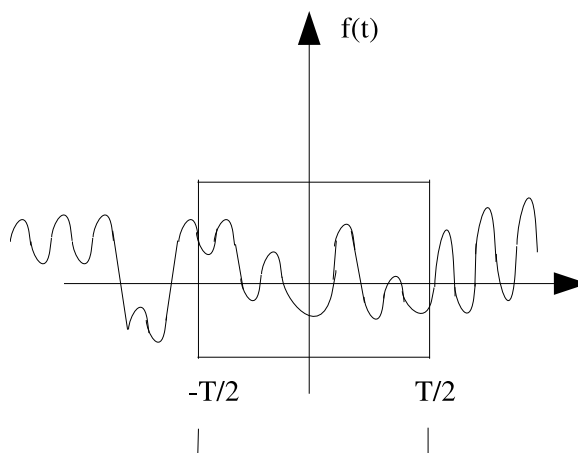


Figure 1.22

1. compute $\frac{\text{Energy}}{T} = \frac{(\|f\|_2)^2}{T}$
2. look at $\lim_{T \rightarrow \infty} \frac{\text{Energy}}{T} = \frac{(\|f\|_2)^2}{T}$

P_f is often called the mean-square value of f . $\sqrt{P_f}$ is called the **root mean squared** (RMS) value of f .
Units?

"Energy signals" have finite norm (energy) $E_f < \infty$.

"Power signals" have finite and nonzero power $P_f < \infty$, $P_f \neq 0$, and ($\rightarrow E_f = \infty$).

1.2.5.1 Conclusions

Energy signals are not power signals.

Power signals are not energy signals.

Why?

Exercise 1.5

Are *all* signals either energy or power signals?

Example 1.9

$f(t) = t$

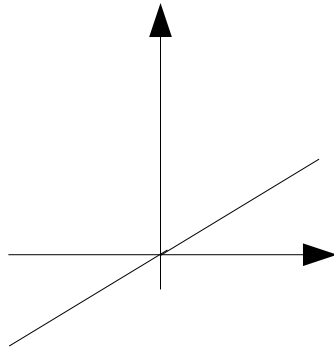


Figure 1.23

The 4 *fundamental classes* of signals we will study depend on the independent (time) variable.

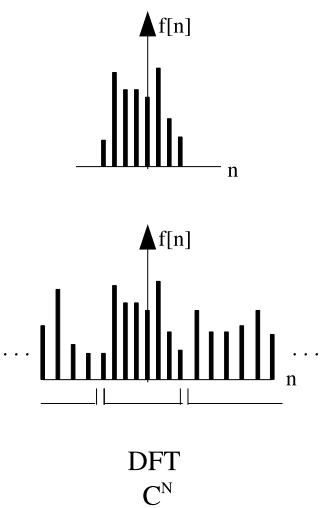
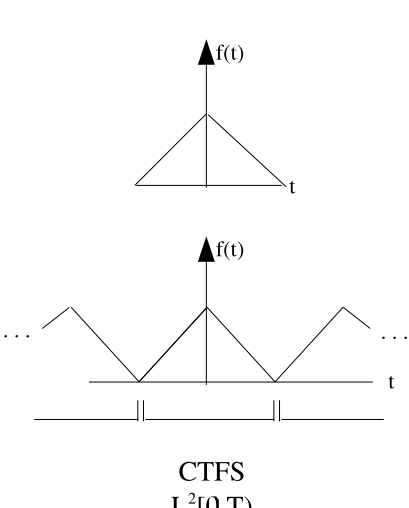
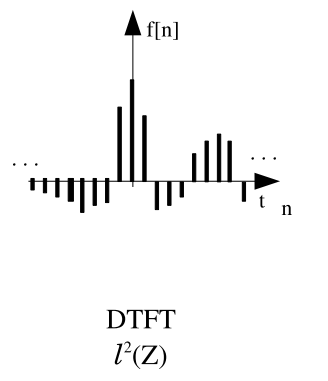
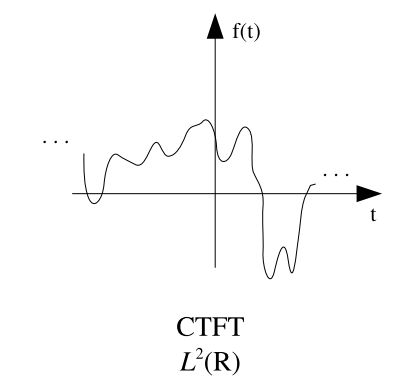
	discrete-time (DT)	continuous-time (CT)
finite-length/ periodic	 <p>DFT C^N</p>	 <p>CTFS $L^2[0,T]$</p>
infinite-length	 <p>DTFT $l^2(Z)$</p>	 <p>CTFT $L^2(\mathbb{R})$</p>

Figure 1.24

1.3 Signal Operations⁷

This module will look at two signal operations, time shifting and time scaling. Signal operations are operations on the time variable of the signal. These operations are very common components to real-world systems and, as such, should be understood thoroughly when learning about signals and systems.

⁷This content is available online at <http://cnx.org/content/m10125/2.8/>.

1.3.1 Time Shifting

Time shifting is, as the name suggests, the shifting of a signal in time. This is done by adding or subtracting the amount of the shift to the time variable in the function. Subtracting a fixed amount from the time variable will shift the signal to the right (delay) that amount, while adding to the time variable will shift the signal to the left (advance).

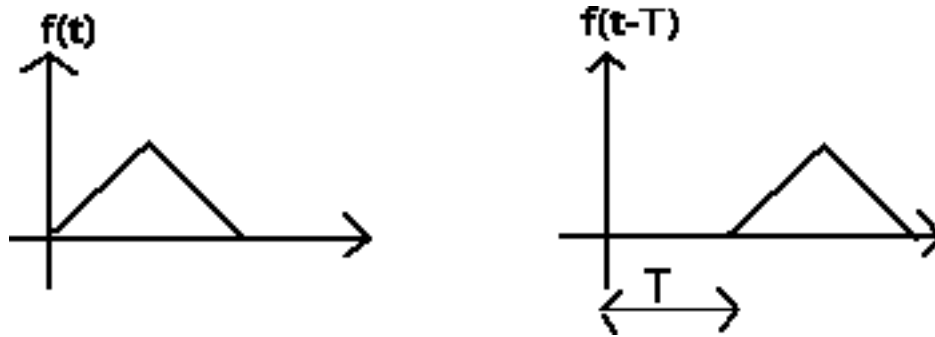


Figure 1.25: $f(t - T)$ moves (delays) f to the right by T .

1.3.2 Time Scaling

Time scaling compresses and dilates a signal by multiplying the time variable by some amount. If that amount is greater than one, the signal becomes narrower and the operation is called compression, while if the amount is less than one, the signal becomes wider and is called dilation. It often takes people quite a while to get comfortable with these operations, as people's intuition is often for the multiplication by an amount greater than one to dilate and less than one to compress.

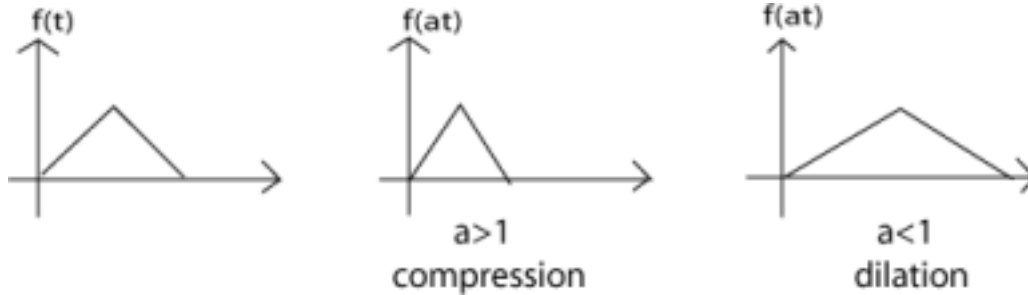


Figure 1.26: $f(at)$ compresses f by a .

Example 1.10

Actually plotting shifted and scaled signals can be quite counter-intuitive. This example will show a fool-proof way to practice this until your proper intuition is developed.

Given $f(t)$, plot $f(-(at))$.

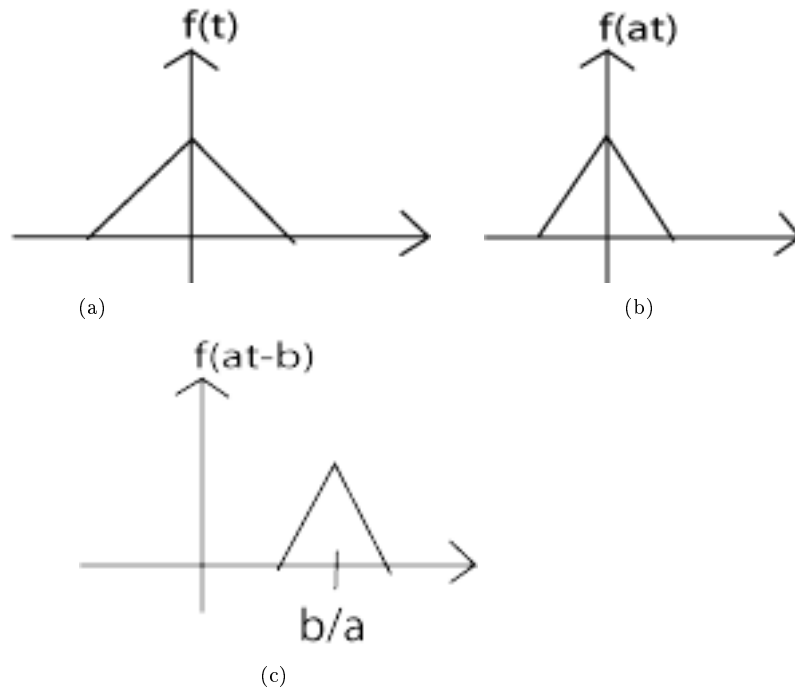


Figure 1.27: (a) Begin with $f(t)$ (b) Then replace t with at to get $f(at)$ (c) Finally, replace t with $t - \frac{b}{a}$ to get $f\left(a\left(t - \frac{b}{a}\right)\right) = f(at - b)$

1.3.3 Time Reversal

A natural question to consider when learning about time scaling is: What happens when the time variable is multiplied by a negative number? The answer to this is time reversal. This operation is the reversal of the time axis, or flipping the signal over the y-axis.

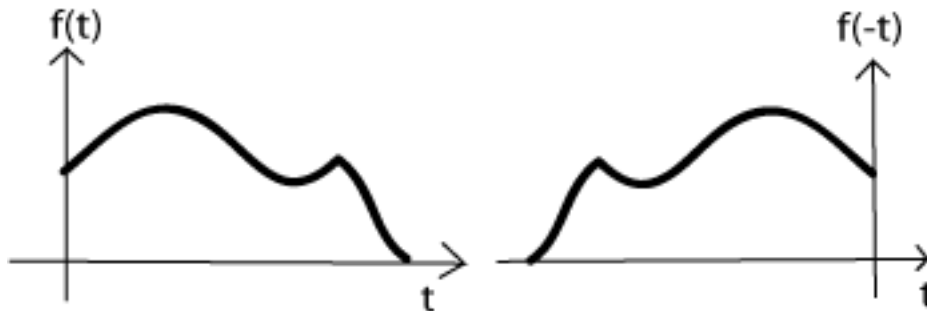


Figure 1.28: Reverse the time axis

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10125/latest/TDSignalOps.llb>

1.4 Useful Signals⁸

Before looking at this module, hopefully you have some basic idea of what a signal is and what basic classifications and properties (Section 1.1) a signal can have. To review, a signal is merely a function defined with respect to an independent variable. This variable is often time but could represent an index of a sequence or any number of things in any number of dimensions. Most, if not all, signals that you will encounter in your studies and the real world will be able to be created from the basic signals we discuss below. Because of this, these elementary signals are often referred to as the *building blocks* for all other signals.

1.4.1 Sinusoids

Probably the most important elemental signal that you will deal with is the real-valued sinusoid. In its continuous-time form, we write the general form as

$$x(t) = A \cos(\omega t + \phi) \quad (1.5)$$

where A is the amplitude, ω is the frequency, and ϕ represents the phase. Note that it is common to see ωt replaced with $2\pi ft$. Since sinusoidal signals are periodic, we can express the period of these, or any periodic signal, as

$$T = \frac{2\pi}{\omega} \quad (1.6)$$

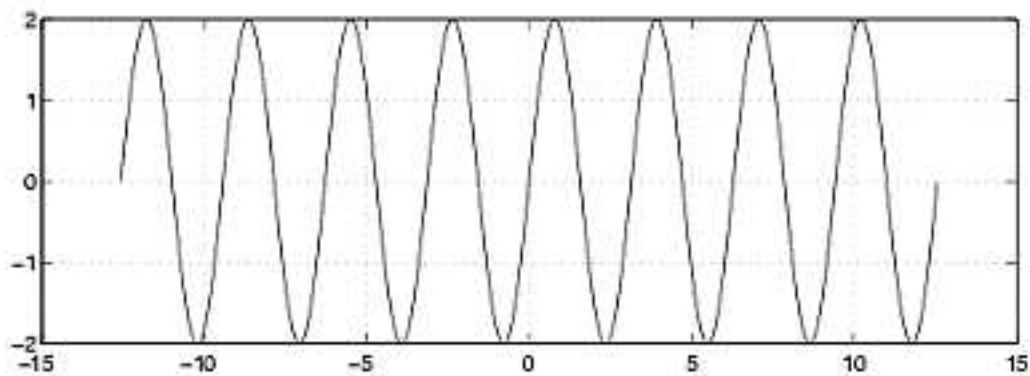


Figure 1.29: Sinusoid with $A = 2$, $w = 2$, and $\phi = 0$.

⁸This content is available online at <<http://cnx.org/content/m10058/2.12/>>.

1.4.2 Complex Exponential Function

Maybe as important as the general sinusoid, the **complex exponential** function will become a critical part of your study of signals and systems. Its general form is written as

$$f(t) = Be^{st} \quad (1.7)$$

where s , shown below, is a complex number in terms of σ , the phase constant, and ω the frequency:

$$s = \sigma + j\omega$$

Please look at the complex exponential module (Section 1.6) or the other elemental signals page⁹ for a much more in depth look at this important signal.

1.4.3 Real Exponentials

Just as the name sounds, real exponentials contain no imaginary numbers and are expressed simply as

$$f(t) = Be^{\alpha t} \quad (1.8)$$

where both B and α are real parameters. Unlike the complex exponential that oscillates, the real exponential either decays or grows depending on the value of α .

- **Decaying Exponential**, when $\alpha < 0$
- **Growing Exponential**, when $\alpha > 0$

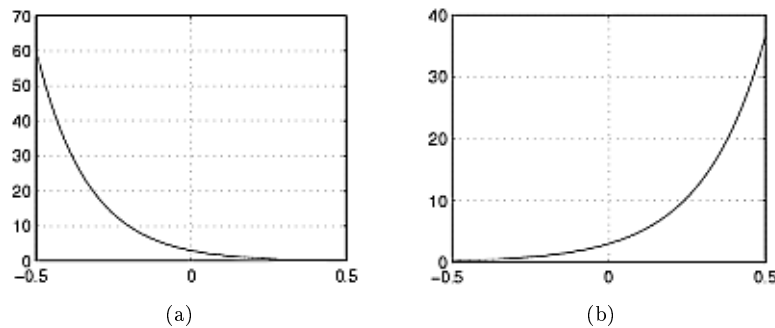


Figure 1.30: Examples of Real Exponentials (a) Decaying Exponential (b) Growing Exponential

1.4.4 Unit Impulse Function

The **unit impulse** (Section 1.5) "function" (or **Dirac delta** function) is a signal that has infinite height and infinitesimal width. However, because of the way it is defined, it actually integrates to one. While in the engineering world, this signal is quite nice and aids in the understanding of many concepts, some mathematicians have a problem with it being called a function, since it is not defined at $t = 0$. Engineers

⁹"Elemental Signals": Section Complex Exponentials <<http://cnx.org/content/m0004/latest/#sec2>>

reconcile this problem by keeping it around integrals, in order to keep it more nicely defined. The unit impulse is most commonly denoted as

$$\delta(t)$$

The most important property of the unit-impulse is shown in the following integral:

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (1.9)$$

1.4.5 Unit-Step Function

Another very basic signal is the **unit-step function** that is defined as

$$u(t) = \begin{cases} 0 & \text{if } t < 0 \\ 1 & \text{if } t \geq 0 \end{cases} \quad (1.10)$$

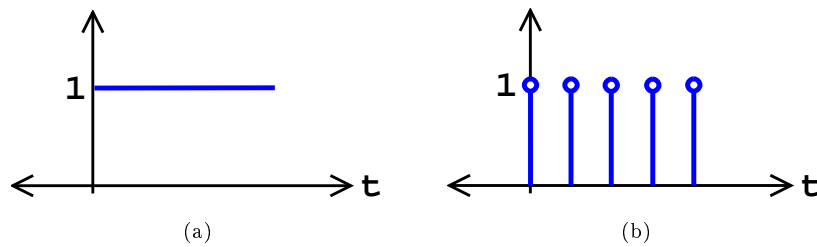


Figure 1.31: Basic Step Functions (a) Continuous-Time Unit-Step Function (b) Discrete-Time Unit-Step Function

Note that the step function is discontinuous at the origin; however, it does not need to be defined here as it does not matter in signal theory. The step function is a useful tool for testing and for defining other signals. For example, when different shifted versions of the step function are multiplied by other signals, one can select a certain portion of the signal and zero out the rest.

1.4.6 Ramp Function

The ramp function is closely related to the unit-step discussed above. Where the unit-step goes from zero to one instantaneously, the ramp function better resembles a real-world signal, where there is some time needed for the signal to increase from zero to its set value, one in this case. We define a ramp function as follows

$$r(t) = \begin{cases} 0 & \text{if } t < 0 \\ \frac{t}{t_0} & \text{if } 0 \leq t \leq t_0 \\ 1 & \text{if } t > t_0 \end{cases} \quad (1.11)$$

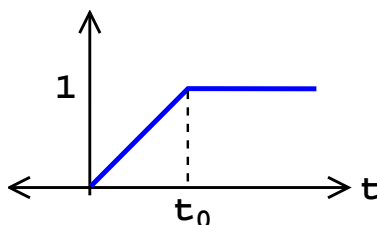


Figure 1.32: Ramp Function

1.5 The Impulse Function¹⁰

In engineering, we often deal with the idea of an action occurring at a point. Whether it be a force at a point in space or a signal at a point in time, it becomes worth while to develop some way of quantitatively defining this. This leads us to the idea of a unit impulse, probably the second most important function, next to the complex exponential (Section 1.6), in systems and signals course.

1.5.1 Dirac Delta Function

The **Dirac Delta function**, often referred to as the unit impulse or delta function, is the function that defines the idea of a unit impulse. This function is one that is infinitesimally narrow, infinitely tall, yet integrates to **unity**, one (see (1.12) below). Perhaps the simplest way to visualize this is as a rectangular pulse from $a - \frac{\epsilon}{2}$ to $a + \frac{\epsilon}{2}$ with a height of $\frac{1}{\epsilon}$. As we take the limit of this, $\lim_{\epsilon \rightarrow 0} 0$, we see that the width tends to zero and the height tends to infinity as the total area remains constant at one. The impulse function is often written as $\delta(t)$.

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (1.12)$$

¹⁰This content is available online at <<http://cnx.org/content/m10059/2.19/>>.

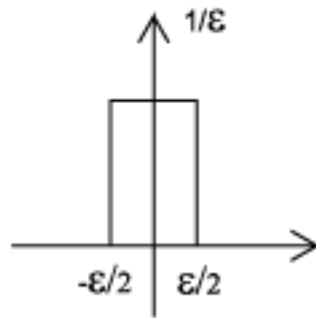


Figure 1.33: This is one way to visualize the Dirac Delta Function.

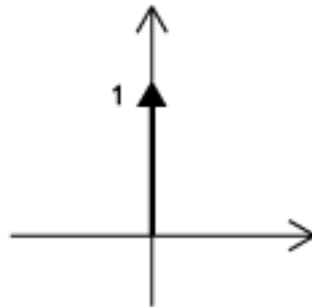


Figure 1.34: Since it is quite difficult to draw something that is infinitely tall, we represent the Dirac with an arrow centered at the point it is applied. If we wish to scale it, we may write the value it is scaled by next to the point of the arrow. This is a unit impulse (no scaling).

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10059/latest/ImpulseFunction.llb>

1.5.1.1 The Sifting Property of the Impulse

The first step to understanding what this unit impulse function gives us is to examine what happens when we multiply another function by it.

$$f(t) \delta(t) = f(0) \delta(t) \quad (1.13)$$

Since the impulse function is zero everywhere except the origin, we essentially just "pick off" the value of the function we are multiplying by evaluated at zero.

At first glance this may not appear to give use much, since we already know that the impulse evaluated at zero is infinity, and anything times infinity is infinity. However, what happens if we integrate this?

Sifting Property

$$\begin{aligned} \int_{-\infty}^{\infty} f(t) \delta(t) dt &= \int_{-\infty}^{\infty} f(0) \delta(t) dt \\ &= f(0) \int_{-\infty}^{\infty} \delta(t) dt \\ &= f(0) \end{aligned} \quad (1.14)$$

It quickly becomes apparent that what we end up with is simply the function evaluated at zero. Had we used $\delta(t - T)$ instead of $\delta(t)$, we could have "sifted out" $f(T)$. This is what we call the **Sifting Property** of the Dirac function, which is often used to define the unit impulse.

The Sifting Property is very useful in developing the idea of convolution (Section 3.2) which is one of the fundamental principles of signal processing. By using convolution and the sifting property we can represent an approximation of any system's output if we know the system's impulse response and input. Click on the convolution link above for more information on this.

1.5.1.2 Other Impulse Properties

Below we will briefly list a few of the other properties of the unit impulse without going into detail of their proofs - we will leave that up to you to verify as most are straightforward. Note that these properties hold for continuous *and* discrete time.

Unit Impulse Properties

- $\delta(\alpha t) = \frac{1}{|\alpha|} \delta(t)$
- $\delta(t) = \delta(-t)$
- $\delta(t) = \frac{d}{dt} u(t)$, where $u(t)$ is the unit step.

1.5.2 Discrete-Time Impulse (Unit Sample)

The extension of the Unit Impulse Function to discrete-time becomes quite trivial. All we really need to realize is that integration in continuous-time equates to summation in discrete-time. Therefore, we are looking for a signal that sums to zero and is zero everywhere except at zero.

Discrete-Time Impulse

$$\delta[n] = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases} \quad (1.15)$$

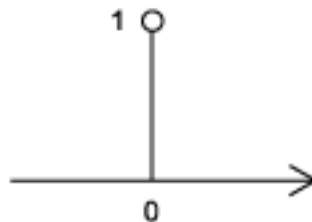


Figure 1.35: The graphical representation of the discrete-time impulse function

Looking at the discrete-time plot of any discrete signal one can notice that all discrete signals are composed of a set of scaled, time-shifted unit samples. If we let the value of a sequence at each integer k be denoted by $s[k]$ and the unit sample delayed that occurs at k to be written as $\delta[n-k]$, we can write any signal as the sum of delayed unit samples that are scaled by the signal value, or weighted coefficients.

$$s[n] = \sum_{k=-\infty}^{\infty} (s[k] \delta[n-k]) \quad (1.16)$$

This decomposition is strictly a property of discrete-time signals and proves to be a very useful property.

NOTE: Through the above reasoning, we have formed (1.16), which is the fundamental concept of discrete-time convolution (Section 4.2).

1.5.3 The Impulse Response

The **impulse response** is exactly what its name implies - the response of an LTI system, such as a filter, when the system's input is the unit impulse (or unit sample). A system can be completely described by its impulse response due to the idea mentioned above that all signals can be represented by a superposition of signals. An impulse response gives an equivalent description of a system as a transfer function¹¹, since they are Laplace Transforms (Section 13.1) of each other.

NOTATION: Most texts use $\delta(t)$ and $\delta[n]$ to denote the continuous-time and discrete-time impulse response, respectively.

1.6 The Complex Exponential¹²

1.6.1 The Exponential Basics

The **complex exponential** is one of the most fundamental and important signal in signal and system analysis. Its importance comes from its functions as a basis for periodic signals as well as being able to characterize linear, time-invariant (Section 2.1) signals. Before proceeding, you should be familiar with the ideas and functions of complex numbers¹³.

1.6.1.1 Basic Exponential

For all numbers x , we easily derive and define the **exponential function** from the Taylor's series below:

$$e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \quad (1.17)$$

$$e^x = \sum_{k=0}^{\infty} \left(\frac{1}{k!} x^k \right) \quad (1.18)$$

We can prove, using the ratio test, that this series does indeed converge. Therefore, we can state that the exponential function shown above is continuous and easily defined.

From this definition, we can prove the following property for exponentials that will be very useful, especially for the complex exponentials discussed in the next section.

$$e^{x_1+x_2} = (e^{x_1})(e^{x_2}) \quad (1.19)$$

¹¹"Transfer Functions" <<http://cnx.org/content/m0028/latest/>>

¹²This content is available online at <<http://cnx.org/content/m10060/2.20/>>.

¹³"Complex Numbers" <<http://cnx.org/content/m0081/latest/>>

1.6.1.2 Complex Continuous-Time Exponential

Now for all complex numbers s , we can define the **complex continuous-time exponential signal** as

$$\begin{aligned} f(t) &= Ae^{st} \\ &= Ae^{j\omega t} \end{aligned} \quad (1.20)$$

where A is a constant, t is our independent variable for time, and for s imaginary, $s = j\omega$. Finally, from this equation we can reveal the ever important **Euler's Identity** (for more information on Euler read this short biography¹⁴):

$$Ae^{j\omega t} = A\cos(\omega t) + j(A\sin(\omega t)) \quad (1.21)$$

From Euler's Identity we can easily break the signal down into its real and imaginary components. Also we can see how exponentials can be combined to represent any real signal. By modifying their frequency and phase, we can represent any signal through a superposition of many signals - all capable of being represented by an exponential.

The above expressions do not include any information on phase however. We can further generalize our above expressions for the exponential to generalize sinusoids with any phase by making a final substitution for s , $s = \sigma + j\omega$, which leads us to

$$\begin{aligned} f(t) &= Ae^{st} \\ &= Ae^{(\sigma + j\omega)t} \\ &= Ae^{\sigma t} e^{j\omega t} \end{aligned} \quad (1.22)$$

where we define S as the **complex amplitude**, or **phasor**, from the first two terms of the above equation as

$$S = Ae^{\sigma t} \quad (1.23)$$

Going back to Euler's Identity, we can rewrite the exponentials as sinusoids, where the phase term becomes much more apparent.

$$f(t) = Ae^{\sigma t} (\cos(\omega t) + j\sin(\omega t)) \quad (1.24)$$

As stated above we can easily break this formula into its real and imaginary part as follows:

$$Re(f(t)) = Ae^{\sigma t} \cos(\omega t) \quad (1.25)$$

$$Im(f(t)) = Ae^{\sigma t} \sin(\omega t) \quad (1.26)$$

1.6.1.3 Complex Discrete-Time Exponential

Finally we have reached the last form of the exponential signal that we will be interested in, the **discrete-time exponential signal**, which we will not give as much detail about as we did for its continuous-time counterpart, because they both follow the same properties and logic discussed above. Because it is discrete, there is only a slightly different notation used to represent its discrete nature

$$\begin{aligned} f[n] &= Be^{snT} \\ &= Be^{j\omega nT} \end{aligned} \quad (1.27)$$

where nT represents the discrete-time instants of our signal.

¹⁴<http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Euler.html>

1.6.2 Euler's Relation

Along with Euler's Identity, Euler also described a way to represent a complex exponential signal in terms of its real and imaginary parts through **Euler's Relation**:

$$\cos(\omega t) = \frac{e^{j\omega t} + e^{-j\omega t}}{2} \quad (1.28)$$

$$\sin(\omega t) = \frac{e^{j\omega t} - e^{-j\omega t}}{2j} \quad (1.29)$$

$$e^{j\omega t} = \cos(\omega t) + j\sin(\omega t) \quad (1.30)$$

1.6.3 Drawing the Complex Exponential

At this point, we have shown how the complex exponential can be broken up into its real part and its imaginary part. It is now worth looking at how we can draw each of these parts. We can see that both the real part and the imaginary part have a sinusoid times a real exponential. We also know that sinusoids oscillate between one and negative one. From this it becomes apparent that the real and imaginary parts of the complex exponential will each oscillate between a window defined by the real exponential part.

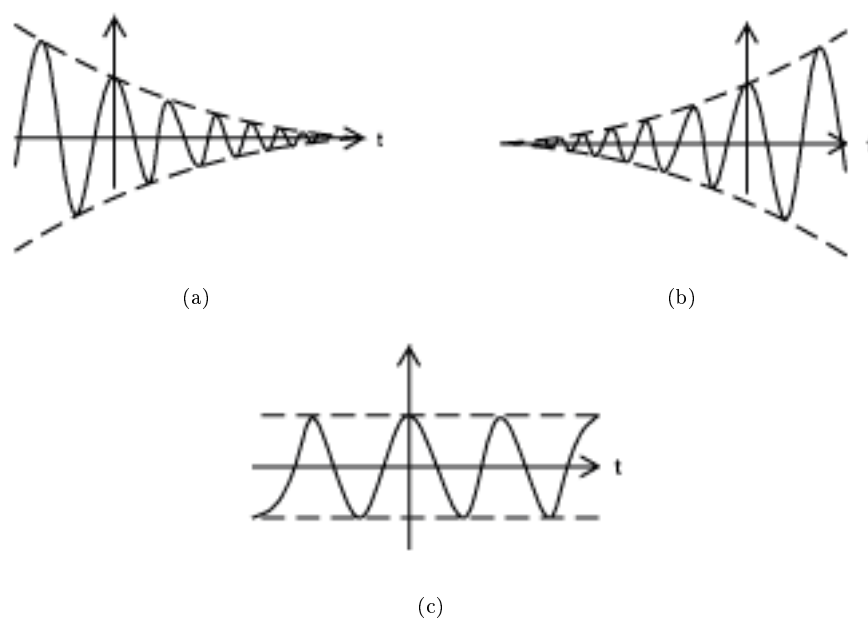


Figure 1.36: The shapes possible for the real part of a complex exponential. Notice that the oscillations are the result of a cosine, as there is a local maximum at $t = 0$. (a) If σ is negative, we have the case of a decaying exponential window. (b) If σ is positive, we have the case of a growing exponential window. (c) If σ is zero, we have the case of a constant window.

While the σ determines the rate of decay/growth, the ω part determines the rate of the oscillations. This is apparent by noticing that the ω is part of the argument to the sinusoidal part.

Exercise 1.6

(Solution on p. 35.)

What do the imaginary parts of the complex exponentials drawn above look like?

Example 1.11

The following demonstration allows you to see how the argument changes the shape of the complex exponential. See here¹⁵ for instructions on how to use the demo.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10060/latest/ComplexEXP.llb>

1.6.4 The Complex Plane

It becomes extremely useful to view the complex variable s as a point in the complex plane¹⁶ (the **s-plane**).

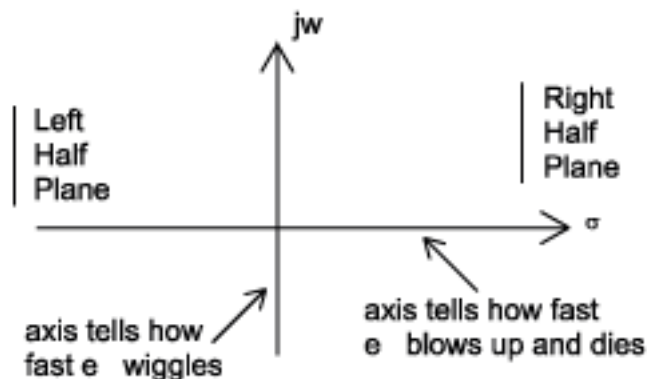


Figure 1.37: This is the s-plane. Notice that any time s lies in the right half plane, the complex exponential will grow through time, while any time it lies in the left half plane it will decay.

1.7 Discrete-Time Signals¹⁷

So far, we have treated what are known as **analog** signals and systems. Mathematically, analog signals are functions having continuous quantities as their independent variables, such as space and time. Discrete-time signals¹⁸ are functions defined on the integers; they are sequences. One of the fundamental results of signal theory¹⁹ will detail conditions under which an analog signal can be converted into a discrete-time one

¹⁵"How to use the LabVIEW demos" <<http://cnx.org/content/m11550/latest/>>

¹⁶"The Complex Plane" <<http://cnx.org/content/m10596/latest/>>

¹⁷This content is available online at <<http://cnx.org/content/m0009/2.23/>>.

¹⁸"Discrete-Time Signals and Systems" <<http://cnx.org/content/m10342/latest/>>

¹⁹"The Sampling Theorem" <<http://cnx.org/content/m0050/latest/>>

and retrieved *without error*. This result is important because discrete-time signals can be manipulated by systems instantiated as computer programs. Subsequent modules describe how virtually all analog signal processing can be performed with software.

As important as such results are, discrete-time signals are more general, encompassing signals derived from analog ones *and* signals that aren't. For example, the characters forming a text file form a sequence, which is also a discrete-time signal. We must deal with such symbolic valued²⁰ signals and systems as well.

As with analog signals, we seek ways of decomposing real-valued discrete-time signals into simpler components. With this approach leading to a better understanding of signal structure, we can exploit that structure to represent information (create ways of representing information with signals) and to extract information (retrieve the information thus represented). For symbolic-valued signals, the approach is different: We develop a common representation of all symbolic-valued signals so that we can embody the information they contain in a unified way. From an information representation perspective, the most important issue becomes, for both real-valued and symbolic-valued signals, efficiency; What is the most parsimonious and compact way to represent information so that it can be extracted later.

1.7.1 Real- and Complex-valued Signals

A discrete-time signal is represented symbolically as $s(n)$, where $n = \{\dots, -1, 0, 1, \dots\}$. We usually draw discrete-time signals as stem plots to emphasize the fact they are functions defined only on the integers. We can delay a discrete-time signal by an integer just as with analog ones. A delayed unit sample has the expression $\delta(n - m)$, and equals one when $n = m$.

Discrete-Time Cosine Signal

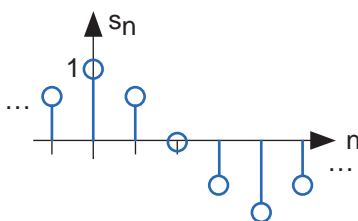


Figure 1.38: The discrete-time cosine signal is plotted as a stem plot. Can you find the formula for this signal?

1.7.2 Complex Exponentials

The most important signal is, of course, the **complex exponential sequence**.

$$s(n) = e^{j2\pi fn} \quad (1.31)$$

1.7.3 Sinusoids

Discrete-time sinusoids have the obvious form $s(n) = A\cos(2\pi fn + \phi)$. As opposed to analog complex exponentials and sinusoids that can have their frequencies be any real value, frequencies of their discrete-time counterparts yield unique waveforms *only* when f lies in the interval $(-\frac{1}{2}, \frac{1}{2}]$. This property can be

²⁰"Discrete-Time Signals and Systems" <<http://cnx.org/content/m10342/latest/#para11>>

easily understood by noting that adding an integer to the frequency of the discrete-time complex exponential has no effect on the signal's value.

$$\begin{aligned} e^{j2\pi(f+m)n} &= e^{j2\pi fn} e^{j2\pi mn} \\ &= e^{j2\pi fn} \end{aligned} \quad (1.32)$$

This derivation follows because the complex exponential evaluated at an integer multiple of 2π equals one.

1.7.4 Unit Sample

The second-most important discrete-time signal is the **unit sample**, which is defined to be

$$\delta(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases} \quad (1.33)$$

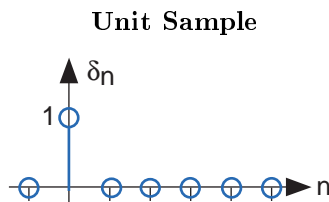


Figure 1.39: The unit sample.

Examination of a discrete-time signal's plot, like that of the cosine signal shown in Figure 1.38 (Discrete-Time Cosine Signal), reveals that all signals consist of a sequence of delayed and scaled unit samples. Because the value of a sequence at each integer m is denoted by $s(m)$ and the unit sample delayed to occur at m is written $\delta(n - m)$, we can decompose *any* signal as a sum of unit samples delayed to the appropriate location and scaled by the signal value.

$$s(n) = \sum_{m=-\infty}^{\infty} (s(m) \delta(n - m)) \quad (1.34)$$

This kind of decomposition is unique to discrete-time signals, and will prove useful subsequently.

Discrete-time systems can act on discrete-time signals in ways similar to those found in analog signals and systems. Because of the role of software in discrete-time systems, many more different systems can be envisioned and “constructed” with programs than can be with analog signals. In fact, a special class of analog signals can be converted into discrete-time signals, processed with software, and converted back into an analog signal, all without the incursion of error. For such signals, systems can be easily produced in software, with equivalent analog realizations difficult, if not impossible, to design.

1.7.5 Symbolic-valued Signals

Another interesting aspect of discrete-time signals is that their values do not need to be real numbers. We do have real-valued discrete-time signals like the sinusoid, but we also have signals that denote the sequence of characters typed on the keyboard. Such characters certainly aren't real numbers, and as a collection of possible signal values, they have little mathematical structure other than that they are members of a set.

More formally, each element of the symbolic-valued signal $s(n)$ takes on one of the values $\{a_1, \dots, a_K\}$ which comprise the **alphabet** A . This technical terminology does not mean we restrict symbols to being members of the English or Greek alphabet. They could represent keyboard characters, bytes (8-bit quantities), integers that convey daily temperature. Whether controlled by software or not, discrete-time systems are ultimately constructed from digital circuits, which consist *entirely* of analog circuit elements. Furthermore, the transmission and reception of discrete-time signals, like e-mail, is accomplished with analog signals and systems. Understanding how discrete-time and analog signals and systems intertwine is perhaps the main goal of this course.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m0009/latest/SignalApprox.llb>

Solutions to Exercises in Chapter 1

Solution to Exercise 1.6 (p. 31)

They look the same except the oscillation is that of a sinusoid as opposed to a cosinusoid (i.e. it passes through the origin rather than being a local maximum at $t = 0$).

Chapter 2

Systems

2.1 System Classifications and Properties¹

2.1.1 Introduction

In this module some of the basic classifications of systems will be briefly introduced and the most important properties of these systems are explained. As can be seen, the properties of a system provide an easy way to separate one system from another. Understanding these basic difference's between systems, and their properties, will be a fundamental concept used in all signal and system courses, such as digital signal processing (DSP). Once a set of systems can be identified as sharing particular properties, one no longer has to deal with proving a certain characteristic of a system each time, but it can simply be accepted do the the systems classification. Also remember that this classification presented here is neither exclusive (systems can belong to several different classifications) nor is it unique (there are other methods of classification ²). Examples of simple systems can be found here³.

2.1.2 Classification of Systems

Along with the classification of systems below, it is also important to understand other Classification of Signals (Section 1.1).

2.1.2.1 Continuous vs. Discrete

This may be the simplest classification to understand as the idea of discrete-time and continuous-time is one of the most fundamental properties to all of signals and system. A system where the input and output signals are continuous is a **continuous system**, and one where the input and output signals are discrete is a **discrete system**.

2.1.2.2 Linear vs. Nonlinear

A **linear** system is any system that obeys the properties of scaling (homogeneity) and superposition (additivity), while a **nonlinear** system is any system that does not obey at least one of these.

To show that a system H obeys the scaling property is to show that

$$H(kf(t)) = kH(f(t)) \quad (2.1)$$

¹This content is available online at <<http://cnx.org/content/m10084/2.19/>>.

²"Introduction to Systems" <<http://cnx.org/content/m0005/latest/>>

³"Simple Systems" <<http://cnx.org/content/m0006/latest/>>

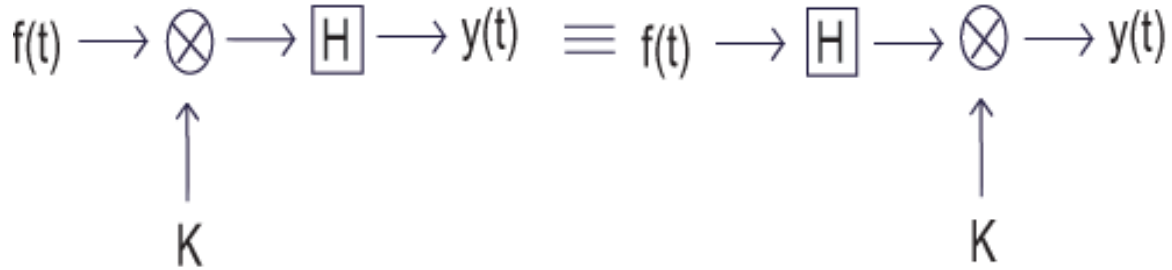


Figure 2.1: A block diagram demonstrating the scaling property of linearity

To demonstrate that a system H obeys the superposition property of linearity is to show that

$$H(f_1(t) + f_2(t)) = H(f_1(t)) + H(f_2(t)) \quad (2.2)$$

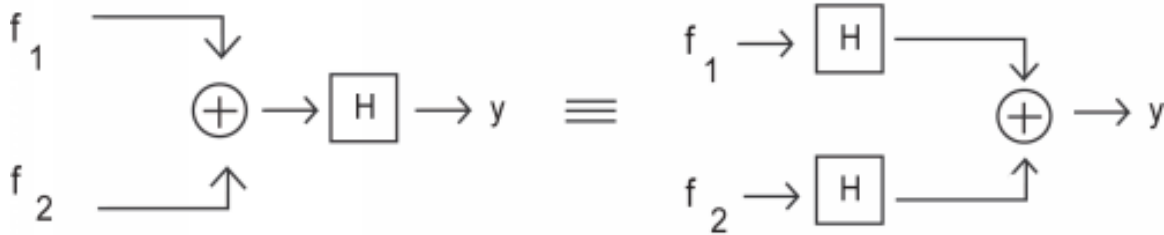


Figure 2.2: A block diagram demonstrating the superposition property of linearity

It is possible to check a system for linearity in a single (though larger) step. To do this, simply combine the first two steps to get

$$H(k_1 f_1(t) + k_2 f_2(t)) = k_1 H(f_1(t)) + k_2 H(f_2(t)) \quad (2.3)$$

2.1.2.3 Time Invariant vs. Time Variant

A **time invariant** system is one that does not depend on when it occurs: the shape of the output does not change with a delay of the input. That is to say that for a system H where $H(f(t)) = y(t)$, H is time invariant if for all T

$$H(f(t - T)) = y(t - T) \quad (2.4)$$

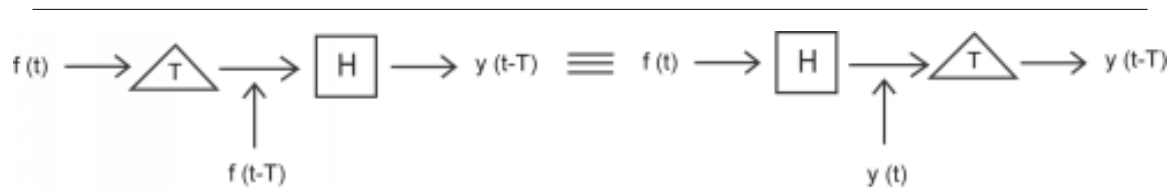


Figure 2.3: This block diagram shows what the condition for time invariance. The output is the same whether the delay is put on the input or the output.

When this property does not hold for a system, then it is said to be **time variant**, or time-varying.

2.1.2.4 Causal vs. Noncausal

A **causal** system is one that is **nonanticipative**; that is, the output may depend on current and past inputs, but not future inputs. All "realtime" systems must be causal, since they can not have future inputs available to them.

One may think the idea of future inputs does not seem to make much physical sense; however, we have only been dealing with time as our dependent variable so far, which is not always the case. Imagine rather that we wanted to do image processing. Then the dependent variable might represent pixels to the left and right (the "future") of the current position on the image, and we would have a **noncausal** system.

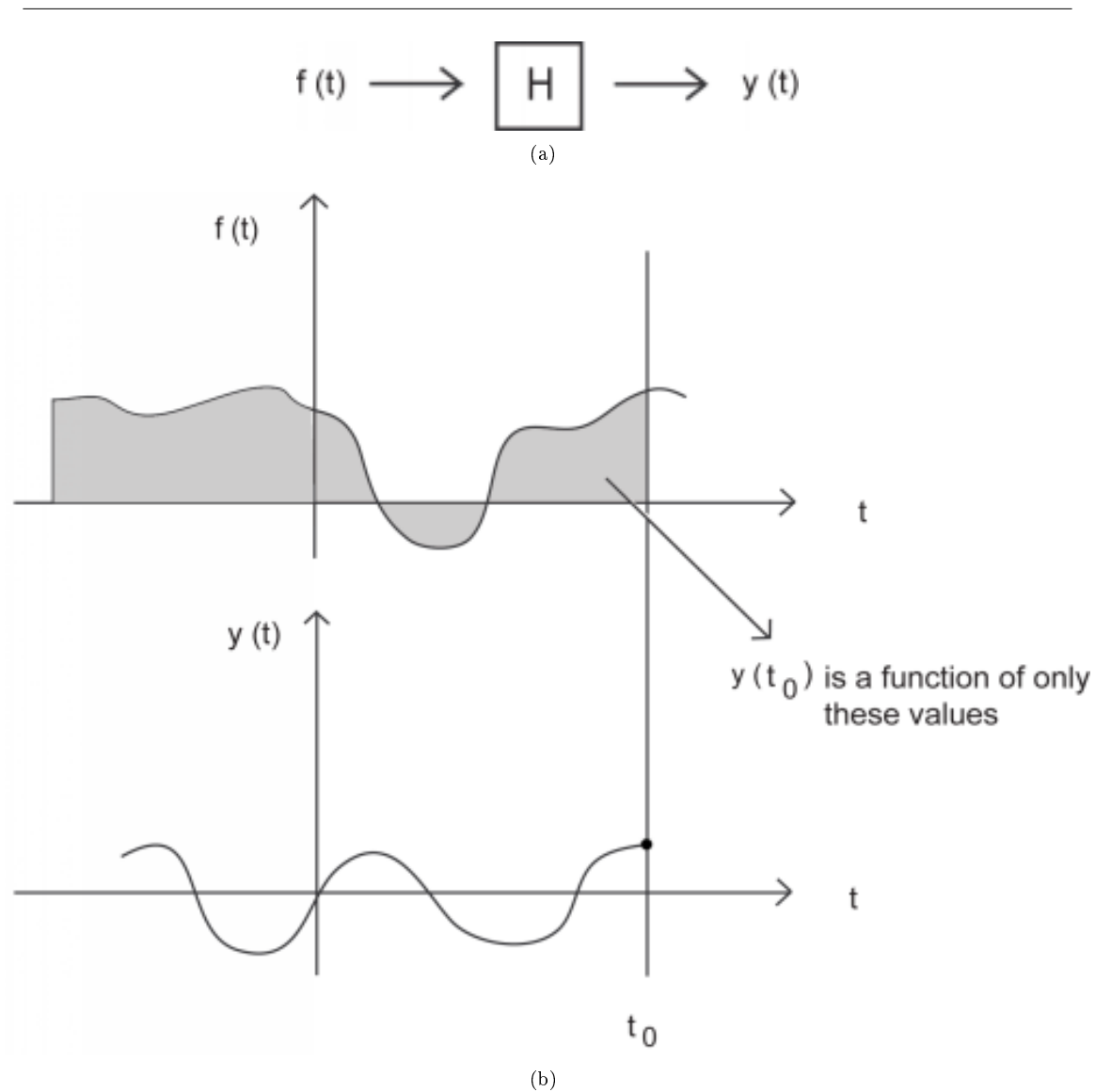


Figure 2.4: (a) For a typical system to be causal... (b) ...the output at time t_0 , $y(t_0)$, can only depend on the portion of the input signal before t_0 .

2.1.2.5 Stable vs. Unstable

A **stable** system is one where the output does not diverge as long as the input does not diverge. There are many ways to say that a signal "diverges"; for example it could have infinite energy. One particularly useful definition of divergence relates to whether the signal is bounded or not. Then a system is referred to as **bounded input-bounded output (BIBO)** stable if *every possible* bounded input produces a bounded output.

Representing this in a mathematical way, a stable system must have the following property, where $x(t)$ is the input and $y(t)$ is the output. The output must satisfy the condition

$$|y(t)| \leq M_y < \infty \quad (2.5)$$

when we have an input to the system that can be described as

$$|x(t)| \leq M_x < \infty \quad (2.6)$$

M_x and M_y both represent a set of finite positive numbers and these relationships hold for all of t .

If these conditions are not met, i.e. a system's output grows without limit (diverges) from a bounded input, then the system is **unstable**. Note that the BIBO stability of a linear time-invariant system (LTI) is neatly described in terms of whether or not its impulse response is absolutely integrable (Section 3.4).

2.2 Properties of Systems⁴

2.2.1 Linear Systems

If a system is linear, this means that when an input to a given system is scaled by a value, the output of the system is scaled by the same amount.

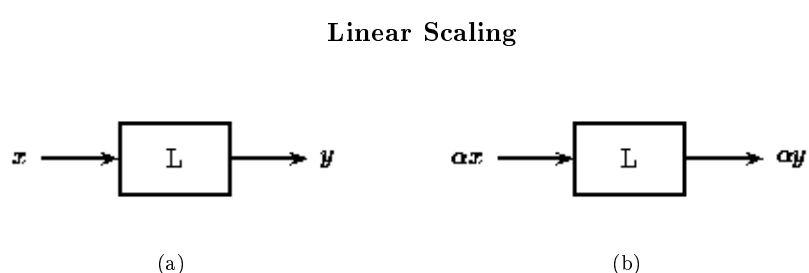


Figure 2.5

In Figure 2.5(a) above, an input x to the linear system L gives the output y . If x is scaled by a value α and passed through this same system, as in Figure 2.5(b), the output will also be scaled by α .

A linear system also obeys the principle of superposition. This means that if two inputs are added together and passed through a linear system, the output will be the sum of the individual inputs' outputs.

⁴This content is available online at <http://cnx.org/content/m2102/2.16/>.

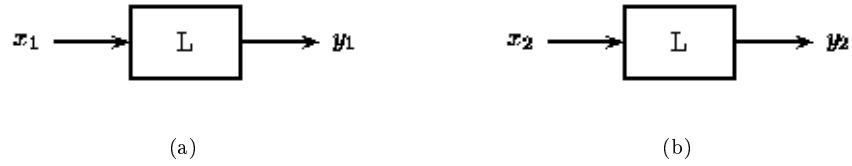


Figure 2.6

Superposition Principle

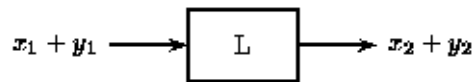


Figure 2.7: If Figure 2.6 is true, then the principle of superposition says that Figure 2.7 (Superposition Principle) is true as well. This holds for linear systems.

That is, if Figure 2.6 is true, then Figure 2.7 (Superposition Principle) is also true for a linear system. The scaling property mentioned above still holds in conjunction with the superposition principle. Therefore, if the inputs x and y are scaled by factors α and β , respectively, then the sum of these scaled inputs will give the sum of the individual scaled outputs:

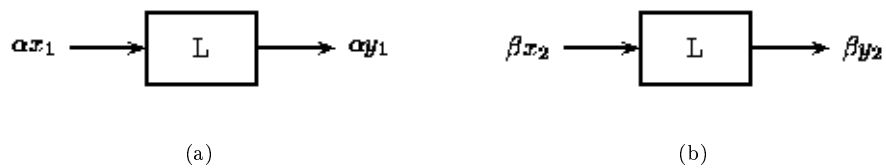


Figure 2.8

Superposition Principle with Linear Scaling

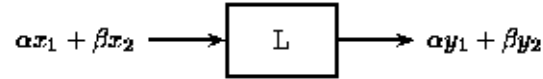


Figure 2.9: Given Figure 2.8 for a linear system, Figure 2.9 (Superposition Principle with Linear Scaling) holds as well.

2.2.2 Time-Invariant Systems

A time-invariant system has the property that a certain input will always give the same output, without regard to when the input was applied to the system.

Time-Invariant Systems

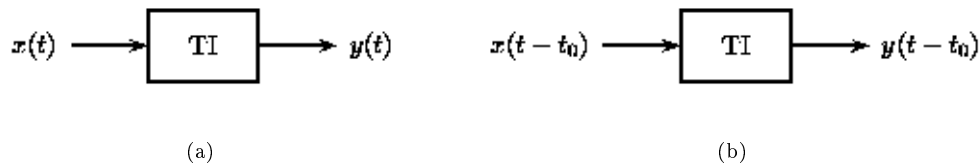


Figure 2.10: Figure 2.10(a) shows an input at time t while Figure 2.10(b) shows the same input t_0 seconds later. In a time-invariant system both outputs would be identical except that the one in Figure 2.10(b) would be delayed by t_0 .

In this figure, $x(t)$ and $x(t - t_0)$ are passed through the system TI. Because the system TI is time-invariant, the inputs $x(t)$ and $x(t - t_0)$ produce the same output. The only difference is that the output due to $x(t - t_0)$ is shifted by a time t_0 .

Whether a system is time-invariant or time-varying can be seen in the differential equation (or difference equation) describing it. *Time-invariant systems are modeled with constant coefficient equations.* A constant coefficient differential (or difference) equation means that the parameters of the system are *not* changing over time and an input now will give the same result as the same input later.

2.2.3 Linear Time-Invariant (LTI) Systems

Certain systems are both linear and time-invariant, and are thus referred to as LTI systems.

Linear Time-Invariant Systems

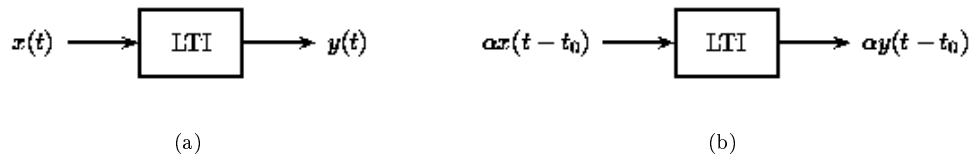


Figure 2.11: This is a combination of the two cases above. Since the input to Figure 2.11(b) is a scaled, time-shifted version of the input in Figure 2.11(a), so is the output.

As LTI systems are a subset of linear systems, they obey the principle of superposition. In the figure below, we see the effect of applying time-invariance to the superposition definition in the linear systems section above.

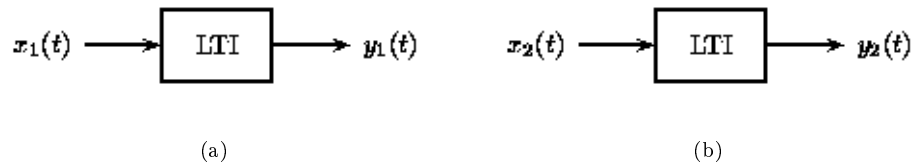


Figure 2.12

Superposition in Linear Time-Invariant Systems

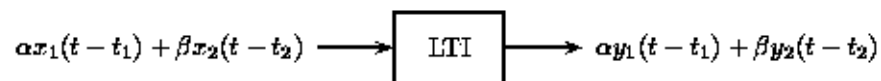
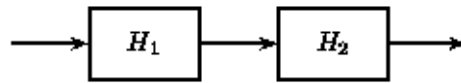


Figure 2.13: The principle of superposition applied to LTI systems

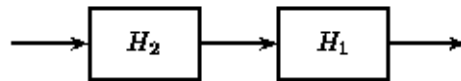
2.2.3.1 LTI Systems in Series

If two or more LTI systems are in series with each other, their order can be interchanged without affecting the overall output of the system. Systems in series are also called cascaded systems.

Cascaded LTI Systems



(a)



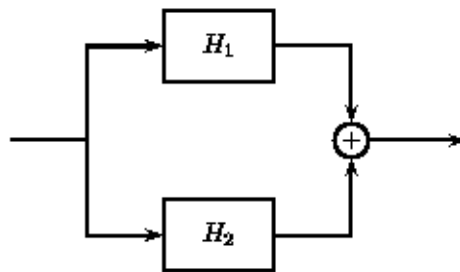
(b)

Figure 2.14: The order of cascaded LTI systems can be interchanged without changing the overall effect.

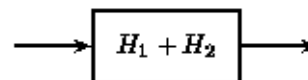
2.2.3.2 LTI Systems in Parallel

If two or more LTI systems are in parallel with one another, an equivalent system is one that is defined as the sum of these individual systems.

Parallel LTI Systems



(a)



(b)

Figure 2.15: Parallel systems can be condensed into the sum of systems.

2.2.4 Causality

A system is causal if it does not depend on future values of the input to determine the output. This means that if the first input to a system comes at time t_0 , then the system should not give any output until that time. An example of a non-causal system would be one that "sensed" an input coming and gave an output before the input arrived:

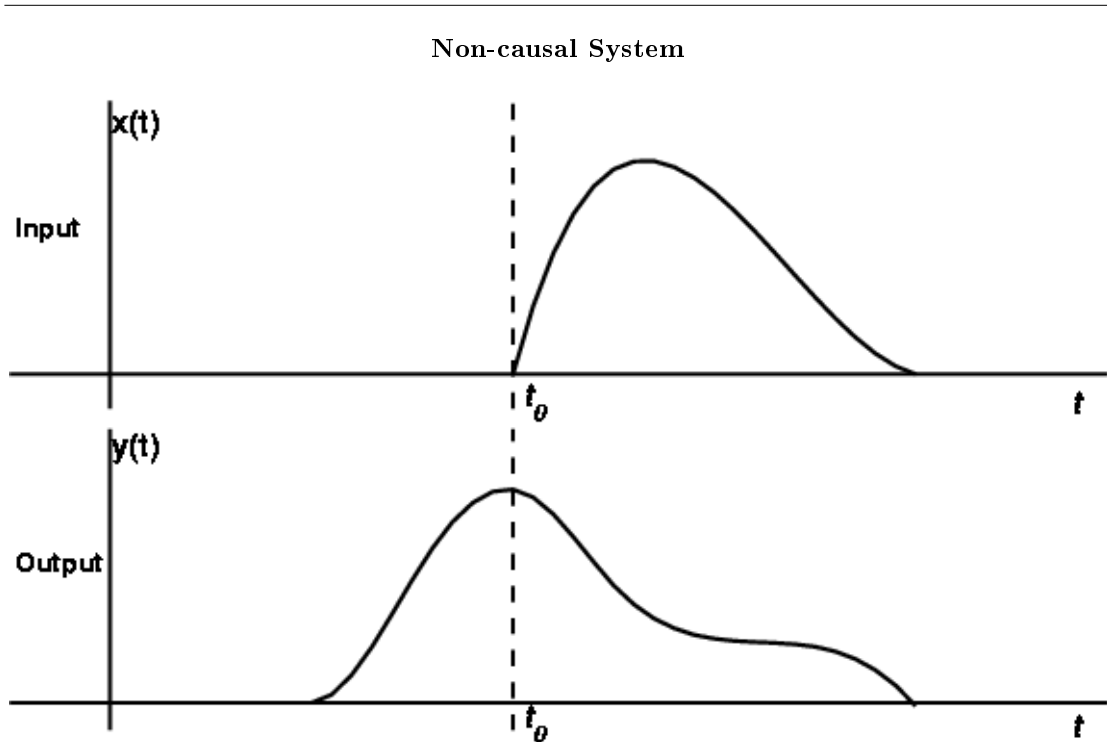


Figure 2.16: In this non-causal system, an output is produced due to an input that occurs later in time.

A causal system is also characterized by an impulse response $h(t)$ that is zero for $t < 0$.

Chapter 3

Time Domain Analysis of Continuous Time Systems

3.1 CT Linear Systems and Differential Equations¹

3.1.1 Continuous-Time Linear Systems

Physically realizable, linear time-invariant systems can be described by a set of linear differential equations (LDEs):

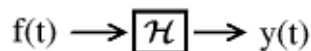


Figure 3.1: Graphical description of a basic linear time-invariant system with an input, $f(t)$ and an output, $y(t)$.

$$\frac{d^n}{dt^n}y(t) + a_{n-1}\frac{d^{n-1}}{dt^{n-1}}y(t) + \cdots + a_1\frac{d}{dt}y(t) + a_0y(t) = b_m\frac{d^m}{dt^m}f(t) + \cdots + b_1\frac{d}{dt}f(t) + b_0f(t)$$

Equivalently,

$$\sum_{i=0}^n \left(a_i \frac{d^i}{dt^i} y(t) \right) = \sum_{i=0}^m \left(b_i \frac{d^i}{dt^i} f(t) \right) \quad (3.1)$$

with $a_n = 1$.

It is easy to show that these equations define a system that is linear and time invariant. A natural question to ask, then, is how to find the system's output response $y(t)$ to an input $f(t)$. Recall that such a solution can be written as

$$y(t) = y_i(t) + y_s(t)$$

We refer to $y_i(t)$ as the **zero-input response** – the homogeneous solution due only to the initial conditions of the system. We refer to $y_s(t)$ as the **zero-state response** – the particular solution in response to the input $f(t)$. We now discuss how to solve for each of these components of the system's response.

¹This content is available online at <<http://cnx.org/content/m10855/2.6/>>.

3.1.1.1 Finding the Zero-Input Response

The zero-input response, $y_i(t)$, is the system response due to initial conditions only.

Example 3.1: Zero-Input Response

Close the switch in the circuit pictured in Figure 3.2 at time $t=0$ and then leave everything else alone. The voltage response is shown in Figure 3.3.

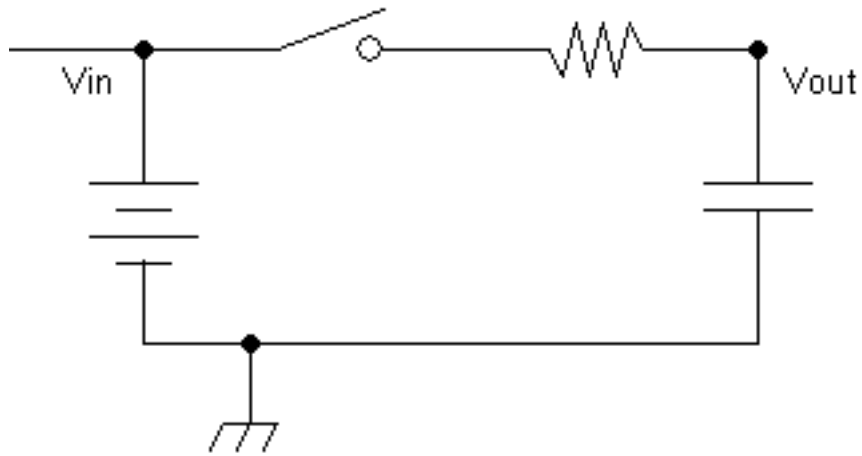


Figure 3.2

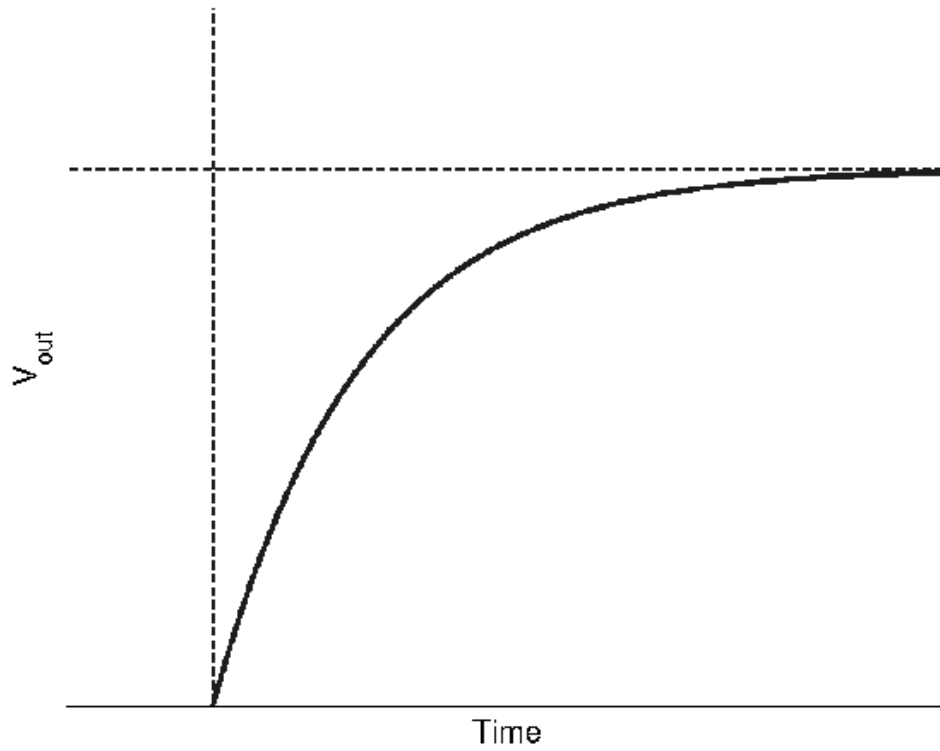


Figure 3.3

Example 3.2: Zero-Input Response

Imagine a mass attached to a spring, as shown in Figure 3.4. When you pull the mass up and let it go, you have an example of a zero-input response. A plot of this response is shown in Figure 3.5.

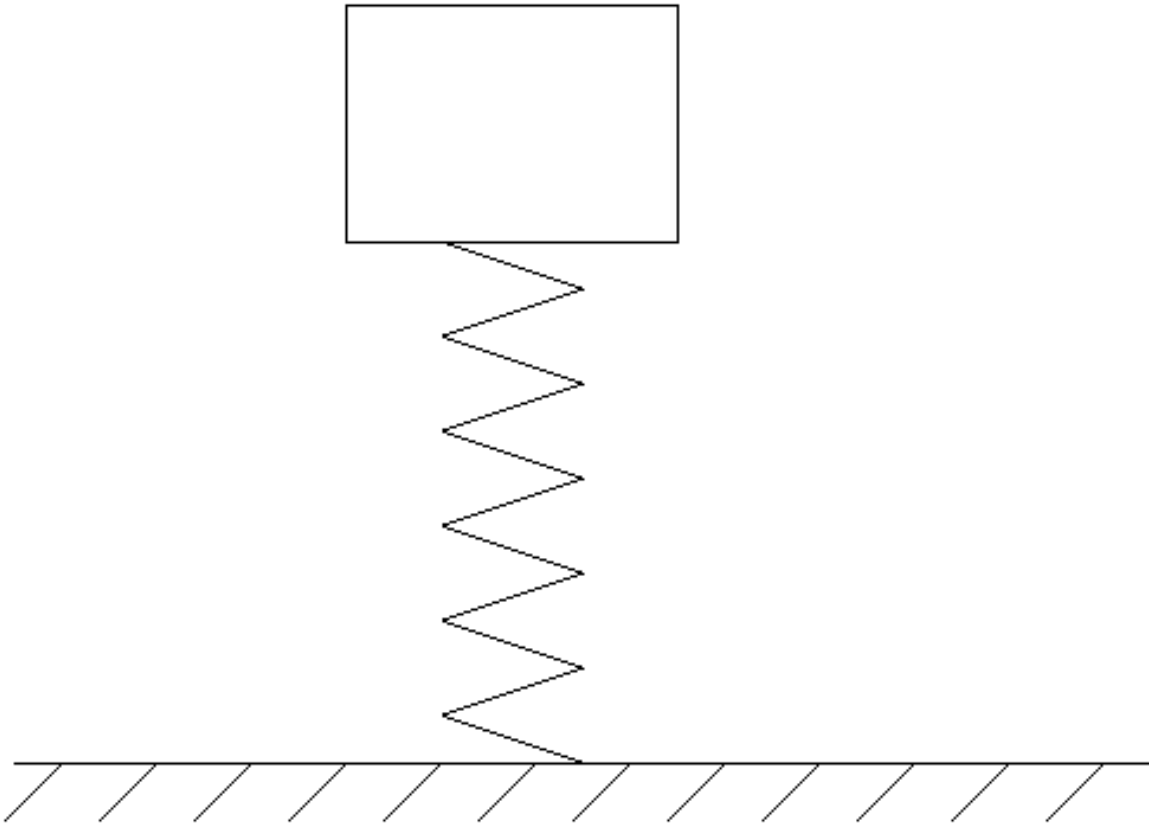


Figure 3.4

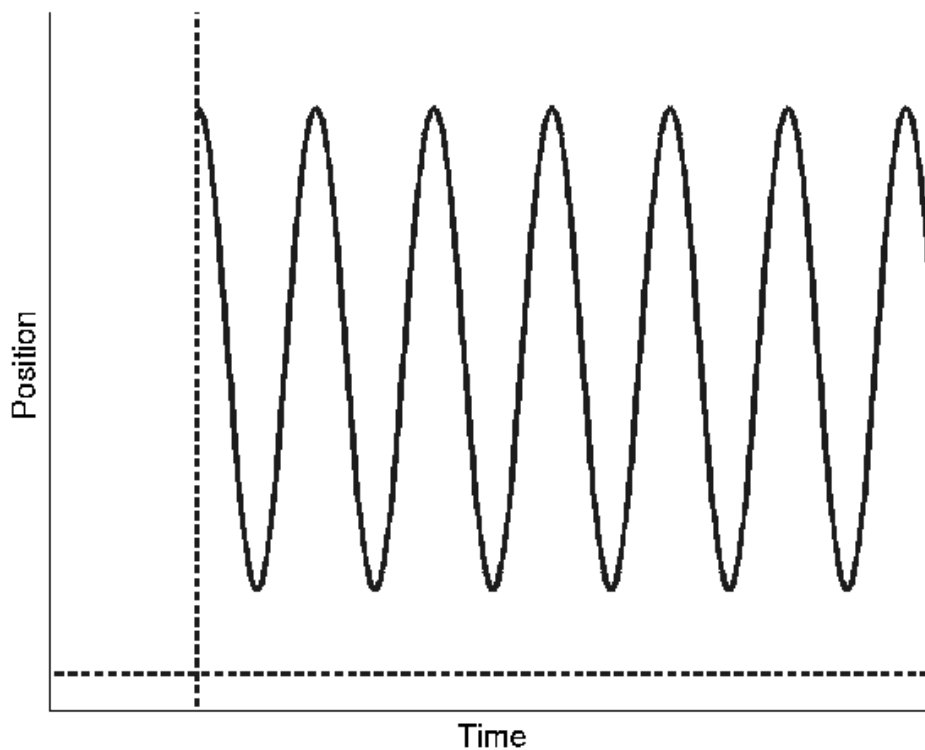


Figure 3.5

There is no input, so we solve for $y_0(t)$ such that

$$\sum_{i=0}^n \left(a_i \frac{d^i}{dt^i} y_0(t) \right) = 0, \quad a_n = 1 \quad (3.2)$$

If D is the derivative operator, we can write the previous equation as:

$$(D^n + a_{n-1}D^{n-1} + \cdots + a_0) y_0(t) = 0 \quad (3.3)$$

Since we need the weighted sum of a bunch of $y_0(t)$'s derivatives to be 0 for all t , then $\left\{ y_0(t), \frac{d}{dt} y_0(t), \frac{d^2}{dt^2} y_0(t), \dots \right\}$ must all be of the same form.

Only the exponential, e^{st} where $s \in \mathbb{C}$, has this property (see your Differential Equation's textbook for details). So we must assume that,

$$y_0(t) = ce^{st}, \quad c \neq 0 \quad (3.4)$$

for some c and s .

Since $\frac{d}{dt} y_0(t) = cse^{st}$, $\frac{d^2}{dt^2} y_0(t) = cs^2e^{st}$, ... we have

$$(D^n + a_{n-1}D^{n-1} + \cdots + a_0) y_0(t) = 0$$

$$c (s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0) e^{st} = 0 \quad (3.5)$$

(3.5) holds for all t only when

$$s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0 = 0 \quad (3.6)$$

Where this equation is referred to as the **characteristic equation** of the system. The possible values of s are the roots of this polynomial $\{s_1, s_2, \dots, s_n\}$

$$(s - s_1)(s - s_2)(s - s_3) \dots (s - s_n) = 0$$

i.e. possible solutions are $c_1e^{s_1t}, c_2e^{s_2t}, \dots, c_ne^{s_nt}$. Since the system is *linear*, the general solution if of the form:

$$y_0(t) = c_1e^{s_1t} + c_2e^{s_2t} + \dots + c_ne^{s_nt} \quad (3.7)$$

Then, solve for the $\{c_1, \dots, c_n\}$ using the initial conditions.

Example 3.3

See Lathi p.108 for a good example!

We generally assume that the IC's of a system are zero, which implies $y_i(t) = 0$. However, the *method* of solving for $y_i(t)$ will prove useful later on.

3.1.1.2 Finding the Zero-State Response

Solving a linear differential equation

$$\sum_{i=0}^n \left(a_i \frac{d^i}{dt^i} y(t) \right) = \sum_{i=0}^m \left(b_i \frac{d^i}{dt^i} f(t) \right) \quad (3.8)$$

given a specific input $f(t)$ is a difficult task in general. More importantly, the method depends entirely on the nature of $f(t)$; if we change the input signal, we must completely re-solve the system of equations to find the system response.

Convolution (Section 3.2) helps to bypass these difficulties. In section 2, we explain how convolution helps to determine the system's output, given only the input $f(t)$ and the system's impulse response (Section 1.5), $h(t)$.

Before deriving the convolution procedure, we show that a system's impulse response is easily derived from its linear, differential equation (LDE). We will show the derivation for the LDE below, where $m < n$:

$$\frac{d^n}{dt^n} y(t) + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} y(t) + \dots + a_1 \frac{d}{dt} y(t) + a_0 y(t) = b_m \frac{d^m}{dt^m} f(t) + \dots + b_1 \frac{d}{dt} f(t) + b_0 f(t) \quad (3.9)$$

We can rewrite (3.9) as

$$Q_D [y(t)] = P_D [f(t)] \quad (3.10)$$

where $Q_D [\cdot]$ is an operator that maps $y(t)$ to the left hand side of (3.9)

$$Q_D [y(t)] = \frac{d^n}{dt^n} y(t) + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} y(t) + \dots + a_1 \frac{d}{dt} y(t) + a_0 y(t) \quad (3.11)$$

and $P_D [\cdot]$ maps $f(t)$ to the right hand side of (3.9). Lathi shows (in Appendix 2.1) that the impulse response of the system described by (3.9) is given by:

$$h(t) = b_n \delta(t) + P_D [y_n(t)] \mu(t) \quad (3.12)$$

where for $m < n$ we have $b_n = 0$. Also, y_n equals the zero input response with initial conditions.

$$\{y^{n-1}(0) = 1, y^{n-2}(0) = 1, \dots, y(0) = 0\}$$

3.2 Continuous-Time Convolution²

3.2.1 Motivation

Convolution helps to determine the effect a system has on an input signal. It can be shown that a linear, time-invariant system (Section 2.1) is completely characterized by its impulse response. At first glance, this may appear to be of little use, since impulse functions are not well defined in real applications. However, the sifting property of impulses (Section 1.5.1.1: The Sifting Property of the Impulse) tells us that a signal can be decomposed into an infinite sum (integral) of scaled and shifted impulses. By knowing how a system affects a single impulse, and by understanding the way a signal is comprised of scaled and summed impulses, it seems reasonable that it should be possible to scale and sum the impulse responses of a system in order to determine what output signal will result from a particular input. This is precisely what convolution does - *convolution determines the system's output from knowledge of the input and the system's impulse response*.

In the rest of this module, we will examine exactly how convolution is defined from the reasoning above. This will result in the convolution integral (see the next section) and its properties (Section 3.3). These concepts are very important in Electrical Engineering and will make any engineer's life a lot easier if the time is spent now to truly understand what is going on.

In order to fully understand convolution, you may find it useful to look at the discrete-time convolution (Section 4.2) as well. It will also be helpful to experiment with the applets³ available on the internet. These resources will offer different approaches to this crucial concept.

3.2.2 Convolution Integral

As mentioned above, the convolution integral provides an easy mathematical way to express the output of an LTI system based on an arbitrary signal, $x(t)$, and the system's impulse response, $h(t)$. The **convolution integral** is expressed as

$$y(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau \quad (3.13)$$

Convolution is such an important tool that it is represented by the symbol $*$, and can be written as

$$y(t) = x(t) * h(t) \quad (3.14)$$

By making a simple change of variables into the convolution integral, $\tau = t - \tau$, we can easily show that convolution is **commutative**:

$$x(t) * h(t) = h(t) * x(t) \quad (3.15)$$

For more information on the characteristics of the convolution integral, read about the Properties of Convolution (Section 3.3).

We now present two distinct approaches for deriving the convolution integral. These derivations, along with a basic example, will help to build intuition about convolution.

3.2.3 Derivation I: The Short Approach

The derivation used here closely follows the one discussed in the Motivation (Section 3.2.1: Motivation) section above. To begin this, it is necessary to state the assumptions we will be making. In this instance, the only constraints on our system are that it be linear and time-invariant.

Brief Overview of Derivation Steps:

1. An impulse input leads to an impulse response output.
2. A shifted impulse input leads to a shifted impulse response output. This is due to the time-invariance of the system.

²This content is available online at <<http://cnx.org/content/m10085/2.26/>>.

³<http://www.jhu.edu/~signals>

3. We now scale the impulse input to get a scaled impulse output. This is using the scalar multiplication property of linearity.
4. We can now "sum up" an infinite number of these scaled impulses to get a sum of an infinite number of scaled impulse responses. This is using the additivity attribute of linearity.
5. Now we recognize that this infinite sum is nothing more than an integral, so we convert both sides into integrals.
6. Recognizing that the input is the function $f(t)$, we also recognize that the output is exactly the convolution integral.

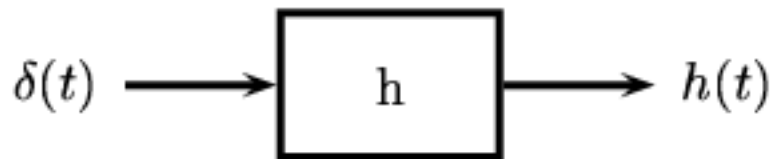


Figure 3.6: We begin with a system defined by its impulse response, $h(t)$.

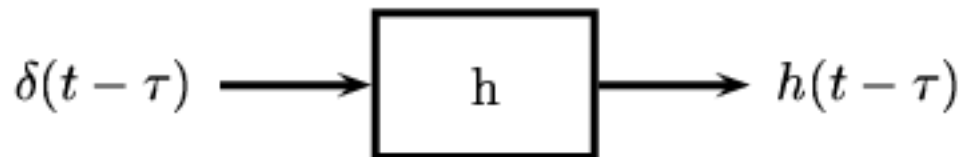


Figure 3.7: We then consider a shifted version of the input impulse. Due to the time invariance of the system, we obtain a shifted version of the output impulse response.

$$\int_{-\infty}^{\infty} f(\tau) \delta(t - \tau) d\tau \longrightarrow \boxed{\text{h}} \longrightarrow \int_{-\infty}^{\infty} f(\tau) h(t - \tau) d\tau$$

Figure 3.8: Now we use the scaling part of linearity by scaling the system by a value, $f(\tau)$, that is constant with respect to the system variable, t .

$$\int_{-\infty}^{\infty} f(\tau) \delta(t - \tau) d\tau \longrightarrow \boxed{\text{h}} \longrightarrow \int_{-\infty}^{\infty} f(\tau) h(t - \tau) d\tau$$

Figure 3.9: We can now use the additivity aspect of linearity to add an infinite number of these, one for each possible τ . Since an infinite sum is exactly an integral, we end up with the integration known as the Convolution Integral. Using the sifting property, we recognize the left-hand side simply as the input $f(t)$.

3.2.4 Derivation II: The Long Approach

This derivation is really not too different from the one above. It is, however, a little more rigorous and a little longer. Hopefully, if you think you "kind of" get the derivation above, this will help you gain a more complete understanding of convolution.

The first step in this derivation is to define a particular realization of the unit impulse function (Section 1.5). For this, we will use $\delta_{\Delta}(t) = \begin{cases} \frac{1}{\Delta} & \text{if } -(\frac{\Delta}{2}) < t < \frac{\Delta}{2} \\ 0 & \text{otherwise} \end{cases}$

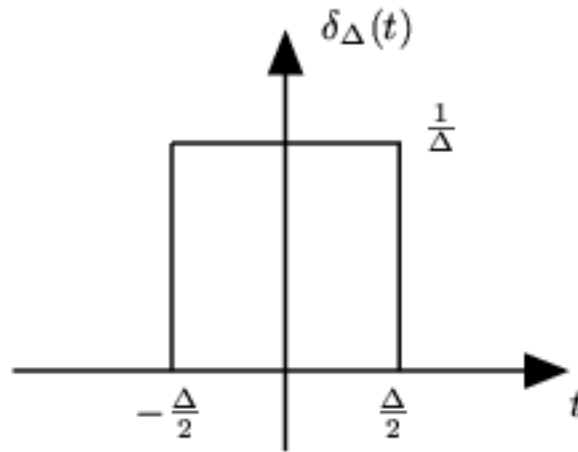


Figure 3.10: The realization of the unit impulse function that we will use for this derivation.

After defining our realization of the unit impulse response, we can derive our convolution integral from the following steps found in the table below. Note that the left column represents the input and the right column is the system's output given that input.

Derivation II of Convolution Integral

Input		Output
$\lim_{\Delta \rightarrow 0} \delta_{\Delta}(t)$	$\rightarrow h \rightarrow$	$\lim_{\Delta \rightarrow 0} h(t)$
$\lim_{\Delta \rightarrow 0} \delta_{\Delta}(t - n\Delta)$	$\rightarrow h \rightarrow$	$\lim_{\Delta \rightarrow 0} h(t - n\Delta)$
$\lim_{\Delta \rightarrow 0} f(n\Delta) \delta_{\Delta}(t - n\Delta) \Delta$	$\rightarrow h \rightarrow$	$\lim_{\Delta \rightarrow 0} f(n\Delta) h(t - n\Delta) \Delta$
$\lim_{\Delta \rightarrow 0} \sum_n (f(n\Delta) \delta_{\Delta}(t - n\Delta) \Delta)$	$\rightarrow h \rightarrow$	$\lim_{\Delta \rightarrow 0} \sum_n (f(n\Delta) h(t - n\Delta) \Delta)$
$\int_{-\infty}^{\infty} f(\tau) \delta(t - \tau) d\tau$	$\rightarrow h \rightarrow$	$\int_{-\infty}^{\infty} f(\tau) h(t - \tau) d\tau$
$f(t)$	$\rightarrow h \rightarrow$	$y(t) = \int_{-\infty}^{\infty} f(\tau) h(t - \tau) d\tau$

3.2.5 Implementation of Convolution

Taking a closer look at the convolution integral, we find that we are multiplying the input signal by the time-reversed impulse response and integrating. This will give us the value of the output at one given value of t . If we then shift the time-reversed impulse response by a small amount, we get the output for another value of t . Repeating this for every possible value of t , yields the total output function. While we would never actually do this computation by hand in this fashion, it does provide us with some insight into what is actually happening. We find that we are essentially reversing the impulse response function and sliding it across the input function, integrating as we go. This method, referred to as the **graphical method**, provides us with a much simpler way to solve for the output for simple (contrived) signals, while improving our intuition for the more complex cases where we rely on computers. In fact Texas Instruments⁴ develops

⁴<http://www.ti.com>

Digital Signal Processors⁵ which have special instruction sets for computations such as convolution.

Example 3.4

This demonstration illustrates the graphical method for convolution. See here⁶ for instructions on how to use the demo.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10085/latest/ConvolutionTime.llb>

3.2.6 Basic Example

Let us look at a basic continuous-time convolution example to help express some of the ideas mentioned above through a short example. We will convolve together two unit pulses, $x(t)$ and $h(t)$.

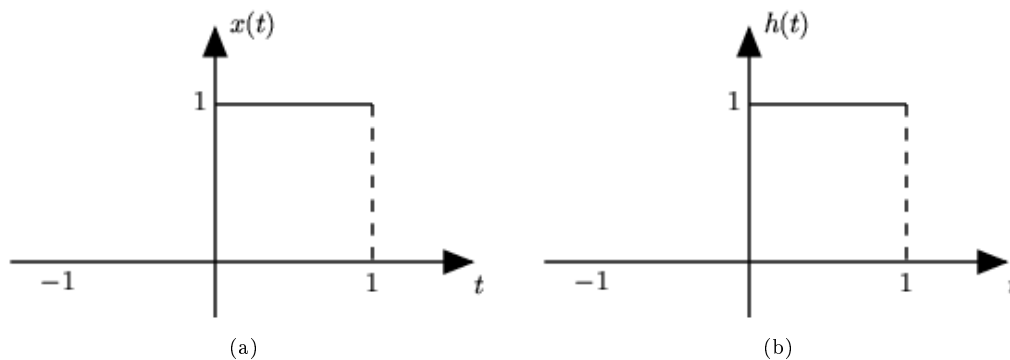


Figure 3.11: Here are the two basic signals that we will convolve together.

3.2.6.1 Reflect and Shift

Now we will take one of the functions and reflect it around the y-axis. Then we must shift the function, such that the origin, the point of the function that was originally on the origin, is labeled as point τ . This step is shown in the figure below, $h(t - \tau)$. Since convolution is commutative it will never matter which function is reflected and shifted; however, as the functions become more complicated reflecting and shifting the "right one" will often make the problem much easier.

⁵<http://dspvillage.ti.com/docs/toolssoftwarehome.jhtml>

⁶"How to use the LabVIEW demos" <<http://cnx.org/content/m11550/latest/>>

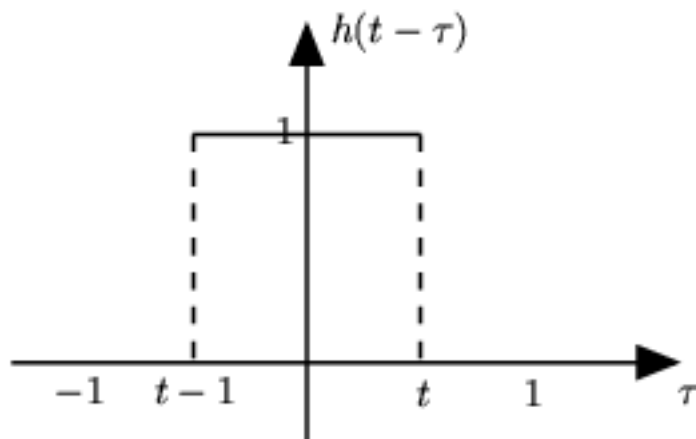


Figure 3.12: The reflected and shifted unit pulse.

3.2.6.2 Regions of Integration

Next, we want to look at the functions and divide the span of the functions into different limits of integration. These different regions can be understood by thinking about how we slide $h(t - \tau)$ over the other function. These limits come from the different regions of overlap that occur between the two functions. If the function were more complex, then we would need to have more limits so that that overlapping parts of both function could be expressed in a single, linear integral. For this problem we will have the following four regions. Compare these limits of integration to the sketches of $h(t - \tau)$ and $x(t)$ to see if you can understand why we have the four regions. Note that the t in the limits of integration refers to the right-hand side of $h(t - \tau)$'s function, labeled as t between zero and one on the plot.

Four Limits of Integration

1. $t < 0$
2. $0 \leq t < 1$
3. $1 \leq t < 2$
4. $t \geq 2$

3.2.6.3 Using the Convolution Integral

Finally we are ready for a little math. Using the convolution integral, let us integrate the product of $x(t)h(t - \tau)$. For our first and fourth region this will be trivial as it will always be 0. The second region, $0 \leq t < 1$, will require the following math:

$$\begin{aligned} y(t) &= \int_0^t 1 d\tau \\ &= t \end{aligned} \tag{3.16}$$

The third region, $1 \leq t < 2$, is solved in much the same manner. Take note of the changes in our integration though. As we move $h(t - \tau)$ across our other function, the left-hand edge of the function, $t - 1$, becomes

our lowlimit for the integral. This is shown through our convolution integral as

$$\begin{aligned}
 y(t) &= \int_{t-1}^1 1d\tau \\
 &= 1 - (t - 1) \\
 &= 2 - t
 \end{aligned}
 \tag{3.17}$$

The above formulas show the method for calculating convolution; however, do not let the simplicity of this example confuse you when you work on other problems. The method will be the same, you will just have to deal with more math in more complicated integrals.

3.2.6.4 Convolution Results

Thus, we have the following results for our four regions:

$$y(t) = \begin{cases} 0 & \text{if } t < 0 \\ t & \text{if } 0 \leq t < 1 \\ 2 - t & \text{if } 1 \leq t < 2 \\ 0 & \text{if } t \geq 2 \end{cases}
 \tag{3.18}$$

Now that we have found the resulting function for each of the four regions, we can combine them together and graph the convolution of $x(t) * h(t)$.

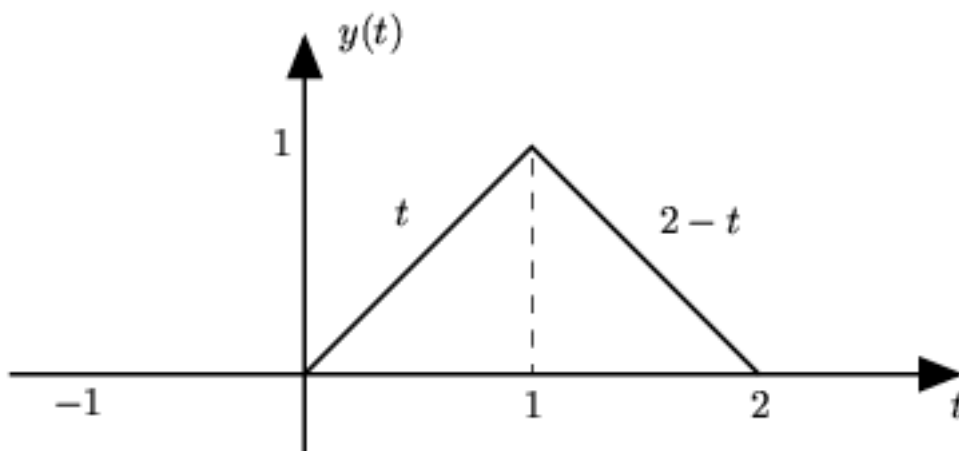


Figure 3.13: Shows the system's response to the input, $x(t)$.

3.3 Properties of Convolution⁷

In this module we will study several of the most prevalent properties of convolution. Note that these properties apply to both continuous-time convolution (Section 3.2) and discrete-time convolution (Section 4.2).

⁷This content is available online at <http://cnx.org/content/m10088/2.14/>.

(Refer back to these two modules if you need a review of convolution). Also, for the proofs of some of the properties, we will be using continuous-time integrals, but we could prove them the same way using the discrete-time summations.

3.3.1 Associativity

Theorem 3.1: Associative Law

$$f_1(t) * (f_2(t) * f_3(t)) = (f_1(t) * f_2(t)) * f_3(t) \quad (3.19)$$

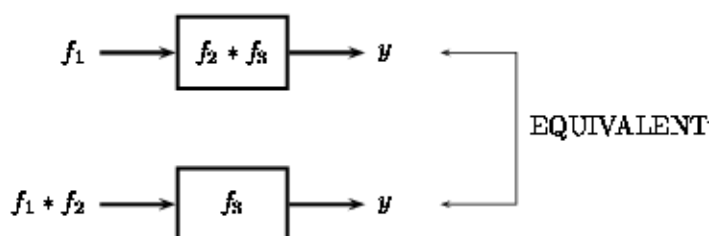


Figure 3.14: Graphical implication of the associative property of convolution.

3.3.2 Commutativity

Theorem 3.2: Commutative Law

$$\begin{aligned} y(t) &= f(t) * h(t) \\ &= h(t) * f(t) \end{aligned} \quad (3.20)$$

Proof: To prove (3.20), all we need to do is make a simple change of variables in our convolution integral (or sum),

$$y(t) = \int_{-\infty}^{\infty} f(\tau) h(t - \tau) d\tau \quad (3.21)$$

By letting $\tau = t - \tau$, we can easily show that convolution is **commutative**:

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} f(t - \tau) h(\tau) d\tau \\ &= \int_{-\infty}^{\infty} h(\tau) f(t - \tau) d\tau \end{aligned} \quad (3.22)$$

$$f(t) * h(t) = h(t) * f(t) \quad (3.23)$$

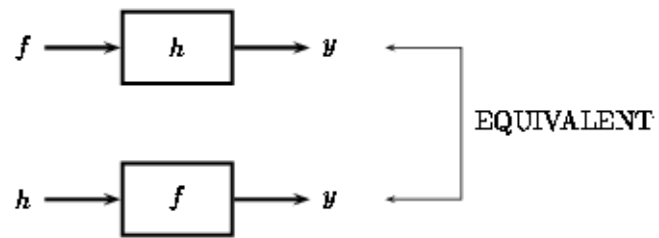


Figure 3.15: The figure shows that either function can be regarded as the system's input while the other is the impulse response.

3.3.3 Distribution

Theorem 3.3: Distributive Law

$$f_1(t) * (f_2(t) + f_3(t)) = f_1(t) * f_2(t) + f_1(t) * f_3(t) \quad (3.24)$$

Proof: The proof of this theorem can be taken directly from the definition of convolution and by using the linearity of the integral.

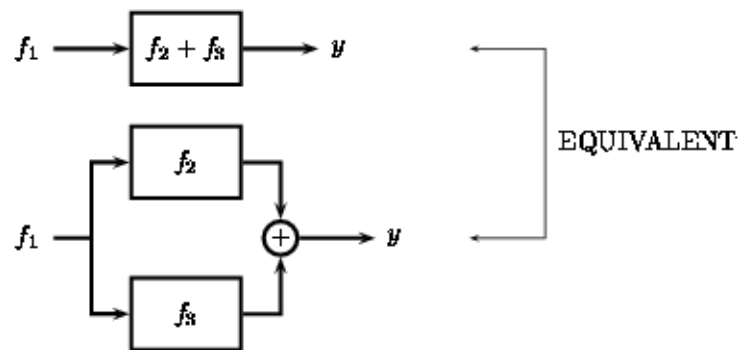


Figure 3.16

3.3.4 Time Shift

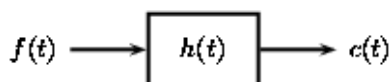
Theorem 3.4: Shift Property

For $c(t) = f(t) * h(t)$, then

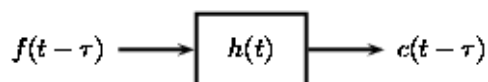
$$c(t - T) = f(t - T) * h(t) \quad (3.25)$$

and

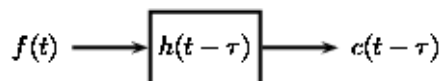
$$c(t - T) = f(t) * h(t - T) \quad (3.26)$$



(a)



(b)



(c)

Figure 3.17: Graphical demonstration of the shift property.

3.3.5 Convolution with an Impulse

Theorem 3.5: Convolving with Unit Impulse

$$f(t) * \delta(t) = f(t) \quad (3.27)$$

Proof: For this proof, we will let $\delta(t)$ be the unit impulse located at the origin. Using the definition of convolution we start with the convolution integral

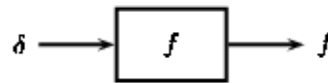
$$f(t) * \delta(t) = \int_{-\infty}^{\infty} \delta(\tau) f(t - \tau) d\tau \quad (3.28)$$

From the definition of the unit impulse, we know that $\delta(\tau) = 0$ whenever $\tau \neq 0$. We use this fact to reduce the above equation to the following:

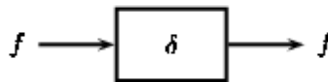
$$\begin{aligned} f(t) * \delta(t) &= \int_{-\infty}^{\infty} \delta(\tau) f(t) d\tau \\ &= f(t) \int_{-\infty}^{\infty} (\delta(\tau)) d\tau \end{aligned} \quad (3.29)$$

The integral of $\delta(\tau)$ will only have a value when $\tau = 0$ (from the definition of the unit impulse), therefore its integral will equal one. Thus we can simplify the equation to our theorem:

$$f(t) * \delta(t) = f(t) \quad (3.30)$$



(a)



(b)

Figure 3.18: The figures, and equation above, reveal the identity function of the unit impulse.

3.3.6 Width

In continuous time, if $Duration(f_1) = T_1$ and $Duration(f_2) = T_2$, then

$$Duration(f_1 * f_2) = T_1 + T_2 \quad (3.31)$$

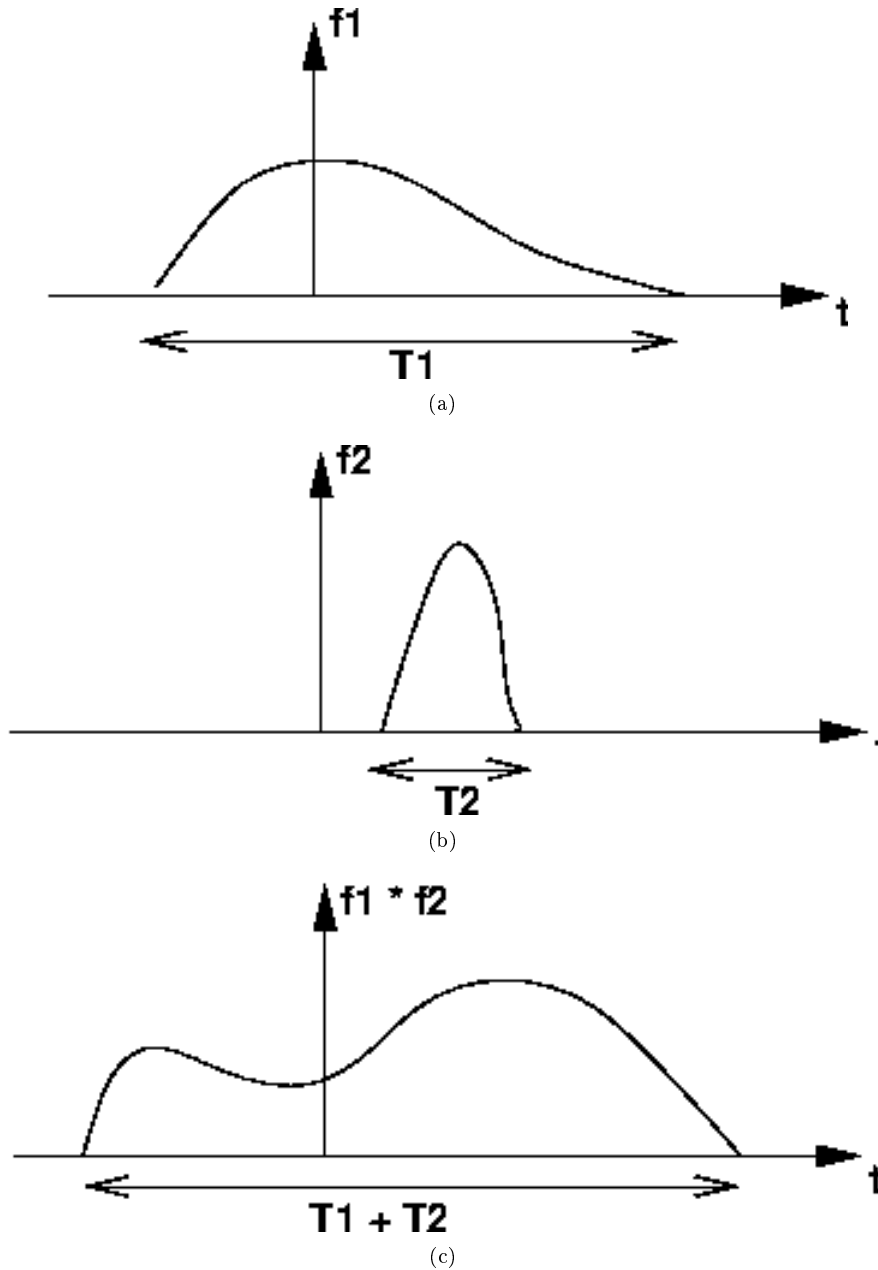


Figure 3.19: In continuous-time, the duration of the convolution result equals the sum of the lengths of each of the two signals that are convolved.

In discrete time, if $\text{Duration}(f_1) = N_1$ and $\text{Duration}(f_2) = N_2$, then

$$\text{Duration}(f_1 * f_2) = N_1 + N_2 - 1 \quad (3.32)$$

3.3.7 Causality

If f and h are both causal, then $f * h$ is also causal.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10088/latest/ConvTIMEDOM.llb>

3.4 BIBO Stability⁸

BIBO stands for bounded input, bounded output. BIBO stable is a condition such that any bounded input yields a bounded output. This is to say that as long as we input a stable signal, we are guaranteed to have a stable output.

In order to understand this concept, we must first look more closely into exactly what we mean by bounded. A bounded signal is any signal such that there exists a value such that the absolute value of the signal is never greater than some value. Since this value is arbitrary, what we mean is that at no point can the signal tend to infinity.

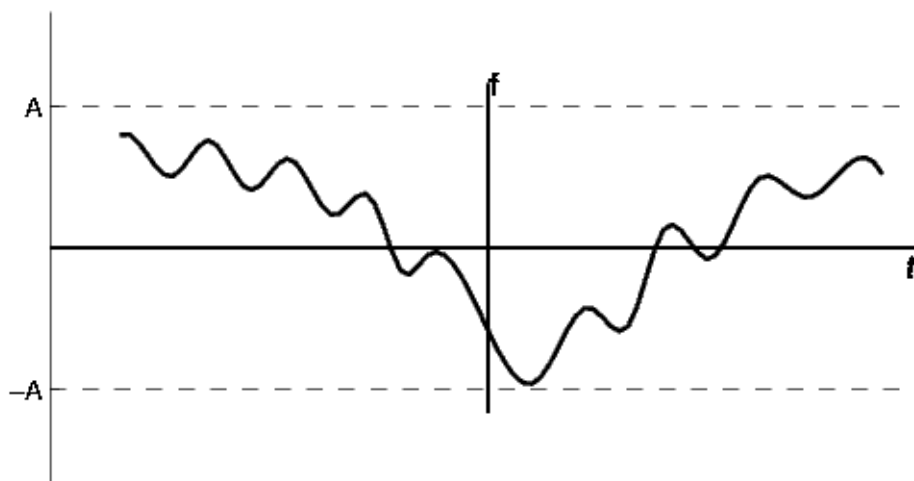


Figure 3.20: A bounded signal is a signal for which there exists a constant A such that $|f(t)| < A$,

Once we have identified what it means for a signal to be bounded, we must turn our attention to the condition a system must possess in order to guarantee that if any bounded signal is passed through the system, a bounded signal will arise on the output. It turns out that a continuous-time LTI (Section 2.1) system with impulse response $h(t)$ is BIBO stable *if and only if*

Continuous-Time Condition for BIBO Stability

$$\int_{-\infty}^{\infty} |h(t)| dt < \infty \quad (3.33)$$

⁸This content is available online at <<http://cnx.org/content/m10113/2.9/>>.

This is to say that the transfer function is *absolutely integrable*.

To extend this concept to discrete-time, we make the standard transition from integration to summation and get that the transfer function, $h(n)$, must be *absolutely summable*. That is

Discrete-Time Condition for BIBO Stability

$$\sum_{n=-\infty}^{\infty} (|h(n)|) < \infty \quad (3.34)$$

3.4.1 Stability and Laplace

Stability is very easy to infer from the pole-zero plot (Section 13.6) of a transfer function. The only condition necessary to demonstrate stability is to show that the $j\omega$ -axis is in the region of convergence.

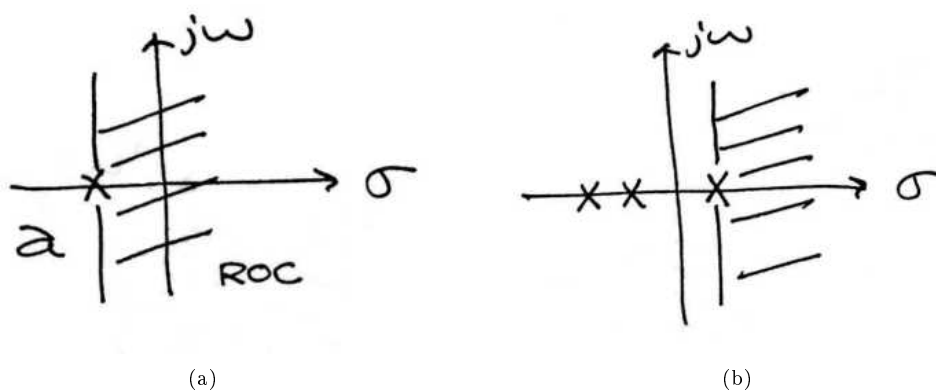


Figure 3.21: (a) Example of a pole-zero plot for a stable continuous-time system. (b) Example of a pole-zero plot for an unstable continuous-time system.

3.4.2 Stability and the Z-Transform

Stability for discrete-time signals (Section 1.1) in the z -domain (Section 14.1) is about as easy to demonstrate as it is for continuous-time signals in the Laplace domain. However, instead of the region of convergence needing to contain the $j\omega$ -axis, the ROC must contain the unit circle.

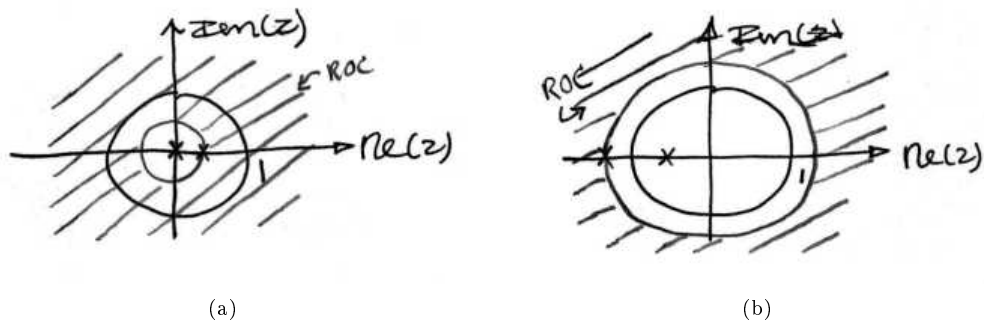


Figure 3.22: (a) A stable discrete-time system. (b) An unstable discrete-time system.

This is an unsupported media type. To view, please see <http://cnx.org/content/m10113/latest/BIBO.llb>

Chapter 4

Time Domain Analysis of Discrete Time Systems

4.1 Discrete-Time Systems in the Time-Domain¹

A discrete-time signal $s(n)$ is *delayed* by n_0 samples when we write $s(n - n_0)$, with $n_0 > 0$. Choosing n_0 to be negative advances the signal along the integers. As opposed to analog delays², discrete-time delays can *only* be integer valued. In the frequency domain, delaying a signal corresponds to a linear phase shift of the signal's discrete-time Fourier transform: $(s(n - n_0) \leftrightarrow e^{-j2\pi f n_0} S(e^{j2\pi f}))$.

Linear discrete-time systems have the superposition property.

$$S(a_1 x_1(n) + a_2 x_2(n)) = a_1 S(x_1(n)) + a_2 S(x_2(n)) \quad (4.1)$$

A discrete-time system is called **shift-invariant** (analogous to time-invariant analog systems³) if delaying the input delays the corresponding output. If $S(x(n)) = y(n)$, then a shift-invariant system has the property

$$S(x(n - n_0)) = y(n - n_0) \quad (4.2)$$

We use the term shift-invariant to emphasize that delays can only have integer values in discrete-time, while in analog signals, delays can be arbitrarily valued.

We want to concentrate on systems that are both linear and shift-invariant. It will be these that allow us the full power of frequency-domain analysis and implementations. Because we have no physical constraints in "constructing" such systems, we need only a mathematical specification. In analog systems, the differential equation specifies the input-output relationship in the time-domain. The corresponding discrete-time specification is the **difference equation**.

$$y(n) = a_1 y(n - 1) + \dots + a_p y(n - p) + b_0 x(n) + b_1 x(n - 1) + \dots + b_q x(n - q) \quad (4.3)$$

Here, the output signal $y(n)$ is related to its *past* values $y(n - l)$, $l = \{1, \dots, p\}$, and to the current and past values of the input signal $x(n)$. The system's characteristics are determined by the choices for the number of coefficients p and q and the coefficients' values $\{a_1, \dots, a_p\}$ and $\{b_0, b_1, \dots, b_q\}$.

ASIDE: There is an asymmetry in the coefficients: where is a_0 ? This coefficient would multiply the $y(n)$ term in (4.3). We have essentially divided the equation by it, which does not change the input-output relationship. We have thus created the convention that a_0 is always one.

¹This content is available online at <<http://cnx.org/content/m10251/2.22/>>.

²"Simple Systems": Section Delay <<http://cnx.org/content/m0006/latest/#delay>>

³"Simple Systems" <<http://cnx.org/content/m0006/latest/#para4wra>>

As opposed to differential equations, which only provide an *implicit* description of a system (we must somehow solve the differential equation), difference equations provide an *explicit* way of computing the output for any input. We simply express the difference equation by a program that calculates each output from the previous output values, and the current and previous inputs.

Difference equations are usually expressed in software with *for* loops. A MATLAB program that would compute the first 1000 values of the output has the form

```
for n=1:1000
    y(n) = sum(a.*y(n-1:-1:n-p)) + sum(b.*x(n:-1:n-q));
end
```

An important detail emerges when we consider making this program work; in fact, as written it has (at least) two bugs. What input and output values enter into the computation of $y(1)$? We need values for $y(0)$, $y(-1)$, ..., values we have not yet computed. To compute them, we would need more previous values of the output, which we have not yet computed. To compute these values, we would need even earlier values, ad infinitum. The way out of this predicament is to specify the system's **initial conditions**: we must provide the p output values that occurred before the input started. These values can be arbitrary, but the choice does impact how the system responds to a given input. *One* choice gives rise to a linear system: Make the initial conditions zero. The reason lies in the definition of a linear system⁴: The only way that the output to a sum of signals can be the sum of the individual outputs occurs when the initial conditions in each case are zero.

Exercise 4.1

(Solution on p. 89.)

The initial condition issue resolves making sense of the difference equation for inputs that start at some index. However, the program will not work because of a programming, not conceptual, error. What is it? How can it be "fixed?"

Example 4.1

Let's consider the simple system having $p = 1$ and $q = 0$.

$$y(n) = ay(n-1) + bx(n) \quad (4.4)$$

To compute the output at some index, this difference equation says we need to know what the previous output $y(n-1)$ and what the input signal is at that moment of time. In more detail, let's compute this system's output to a unit-sample input: $x(n) = \delta(n)$. Because the input is zero for negative indices, we start by trying to compute the output at $n = 0$.

$$y(0) = ay(-1) + b \quad (4.5)$$

What is the value of $y(-1)$? Because we have used an input that is zero for all negative indices, it is reasonable to assume that the output is also zero. Certainly, the difference equation would not describe a linear system⁵ if the input that is zero for *all* time did not produce a zero output. With this assumption, $y(-1) = 0$, leaving $y(0) = b$. For $n > 0$, the input unit-sample is zero, which leaves us with the difference equation $y(n) = ay(n-1)$, $n > 0$. We can envision how the filter responds to this input by making a table.

$$y(n) = ay(n-1) + b\delta(n) \quad (4.6)$$

⁴"Simple Systems": Section Linear Systems <<http://cnx.org/content/m0006/latest/#linearsys>>

⁵"Simple Systems": Section Linear Systems <<http://cnx.org/content/m0006/latest/#linearsys>>

n	$x(n)$	$y(n)$
-1	0	0
0	1	b
1	0	ba
2	0	ba^2
:	0	:
n	0	ba^n

Figure 4.1

Coefficient values determine how the output behaves. The parameter b can be any value, and serves as a gain. The effect of the parameter a is more complicated (Figure 4.1). If it equals zero, the output simply equals the input times the gain b . For all non-zero values of a , the output lasts forever; such systems are said to be **IIR** (*Infinite Impulse Response*). The reason for this terminology is that the unit sample also known as the impulse (especially in analog situations), and the system's response to the "impulse" lasts forever. If a is positive and less than one, the output is a decaying exponential. When $a = 1$, the output is a unit step. If a is negative and greater than -1 , the output oscillates while decaying exponentially. When $a = -1$, the output changes sign forever, alternating between b and $-b$. More dramatic effects when $|a| > 1$; whether positive or negative, the output signal becomes larger and larger, *growing* exponentially.

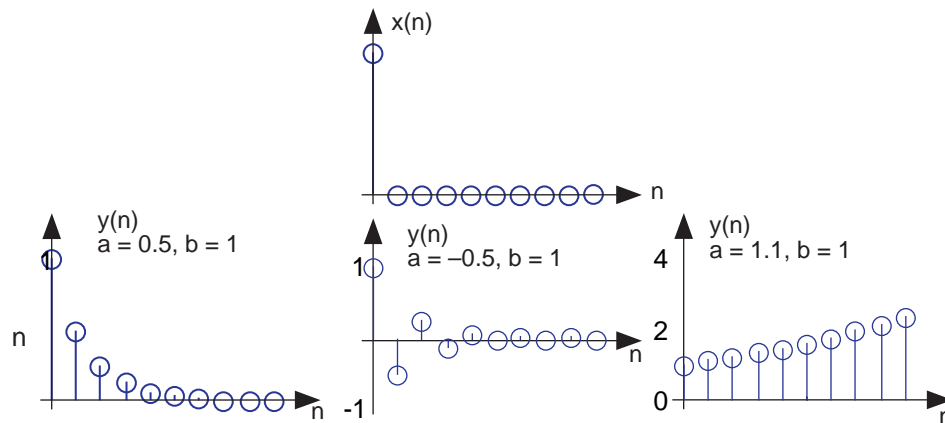


Figure 4.2: The input to the simple example system, a unit sample, is shown at the top, with the outputs for several system parameter values shown below.

Positive values of a are used in population models to describe how population size increases over time. Here, n might correspond to generation. The difference equation says that the number in the next generation is some multiple of the previous one. If this multiple is less than one, the population becomes extinct; if greater than one, the population flourishes. The same difference equation also describes the effect of compound interest on deposits. Here, n indexes the times at which compounding occurs (daily, monthly, etc.), a equals the compound interest rate plus one, and $b = 1$ (the bank provides no gain). In signal processing applications, we typically require that the

output remain bounded for any input. For our example, that means that we restrict $|a| = 1$ and chose values for it and the gain according to the application.

Exercise 4.2

(Solution on p. 89.)

Note that the difference equation (4.3),

$$y(n) = a_1y(n-1) + \cdots + a_py(n-p) + b_0x(n) + b_1x(n-1) + \cdots + b_qx(n-q)$$

does not involve terms like $y(n+1)$ or $x(n+1)$ on the equation's right side. Can such terms also be included? Why or why not?

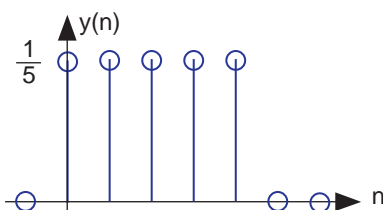


Figure 4.3: The plot shows the unit-sample response of a length-5 boxcar filter.

Example 4.2

A somewhat different system has no "a" coefficients. Consider the difference equation

$$y(n) = \frac{1}{q}(x(n) + \cdots + x(n-q+1)) \quad (4.7)$$

Because this system's output depends only on current and previous input values, we need not be concerned with initial conditions. When the input is a unit-sample, the output equals $\frac{1}{q}$ for $n = \{0, \dots, q-1\}$, then equals zero thereafter. Such systems are said to be **FIR** (Finite Impulse Response) because their unit sample responses have finite duration. Plotting this response (Figure 4.3) shows that the unit-sample response is a pulse of width q and height $\frac{1}{q}$. This waveform is also known as a boxcar, hence the name **boxcar filter** given to this system. We'll derive its frequency response and develop its filtering interpretation in the next section. For now, note that the difference equation says that each output value equals the average of the input's current and previous values. Thus, the output equals the running average of input's previous q values. Such a system could be used to produce the average weekly temperature ($q = 7$) that could be updated daily.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10251/latest/DiscreteTimeSys.llb>

4.2 Discrete-Time Convolution⁶

4.2.1 Overview

Convolution is a concept that extends to all systems that are both **linear and time-invariant (Section 2.1) (LTI)**. The idea of **discrete-time convolution** is exactly the same as that of continuous-time convolution (Section 3.2). For this reason, it may be useful to look at both versions to help your understanding of this extremely important concept. Recall that convolution is a very powerful tool in determining a system's output from knowledge of an arbitrary input and the system's impulse response. It will also be helpful to see convolution graphically with your own eyes and to play around with it some, so experiment with the applets⁷ available on the internet. These resources will offer different approaches to this crucial concept.

4.2.2 Convolution Sum

As mentioned above, the convolution sum provides a concise, mathematical way to express the output of an LTI system based on an arbitrary discrete-time input signal and the system's response. The **convolution sum** is expressed as

$$y[n] = \sum_{k=-\infty}^{\infty} (x[k] h[n-k]) \quad (4.8)$$

As with continuous-time, convolution is represented by the symbol $*$, and can be written as

$$y[n] = x[n] * h[n] \quad (4.9)$$

By making a simple change of variables into the convolution sum, $k = n - k$, we can easily show that convolution is **commutative**:

$$x[n] * h[n] = h[n] * x[n] \quad (4.10)$$

For more information on the characteristics of convolution, read about the Properties of Convolution (Section 3.3).

4.2.3 Derivation

We know that any discrete-time signal can be represented by a summation of scaled and shifted discrete-time impulses. Since we are assuming the system to be linear and time-invariant, it would seem to reason that an input signal comprised of the sum of scaled and shifted impulses would give rise to an output comprised of a sum of scaled and shifted impulse responses. This is exactly what occurs in **convolution**. Below we present a more rigorous and mathematical look at the derivation:

Letting \mathcal{H} be a DT LTI system, we start with the following equation and work our way down the convolution sum!

$$\begin{aligned} y[n] &= \mathcal{H}[x[n]] \\ &= \mathcal{H}\left[\sum_{k=-\infty}^{\infty} (x[k] \delta[n-k])\right] \\ &= \sum_{k=-\infty}^{\infty} (\mathcal{H}[x[k] \delta[n-k]]) \\ &= \sum_{k=-\infty}^{\infty} (x[k] \mathcal{H}[\delta[n-k]]) \\ &= \sum_{k=-\infty}^{\infty} (x[k] h[n-k]) \end{aligned} \quad (4.11)$$

Let us take a quick look at the steps taken in the above derivation. After our initial equation, we using the DT sifting property (Section 1.5.1.1: The Sifting Property of the Impulse) to rewrite the function, $x[n]$, as a sum of the function times the unit impulse. Next, we can move around the \mathcal{H} operator and the summation because $\mathcal{H}[\cdot]$ is a linear, DT system. Because of this linearity and the fact that $x[k]$ is a constant, we can

⁶This content is available online at <<http://cnx.org/content/m10087/2.18/>>.

⁷<http://www.jhu.edu/~signals>

pull the previously mentioned constant out and simply multiply it by $\mathcal{H}[\cdot]$. Finally, we use the fact that $\mathcal{H}[\cdot]$ is time invariant in order to reach our final state - the convolution sum!

A quick graphical example may help in demonstrating why convolution works.

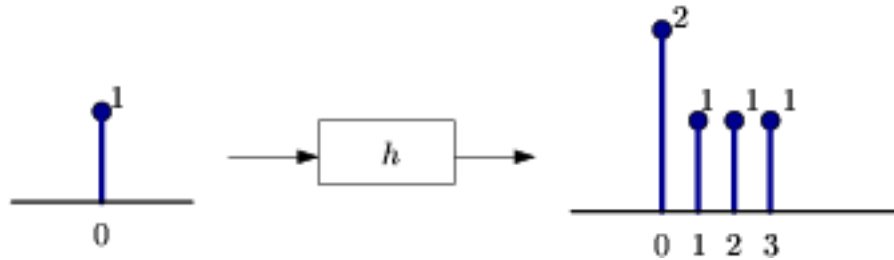


Figure 4.4: A single impulse input yields the system's impulse response.

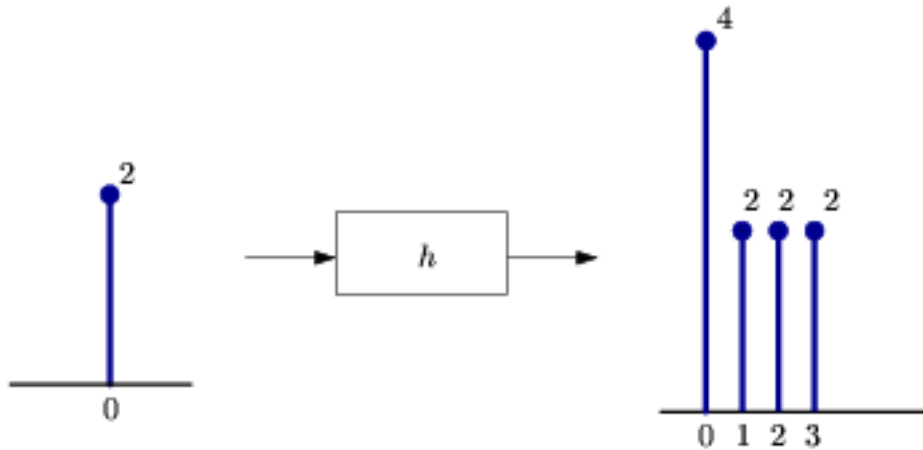


Figure 4.5: A scaled impulse input yields a scaled response, due to the scaling property of the system's linearity.

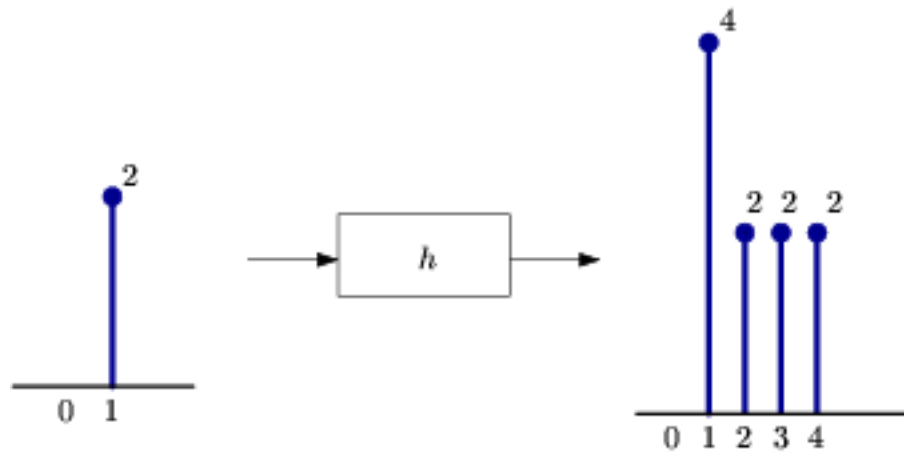


Figure 4.6: We now use the time-invariance property of the system to show that a delayed input results in an output of the same shape, only delayed by the same amount as the input.

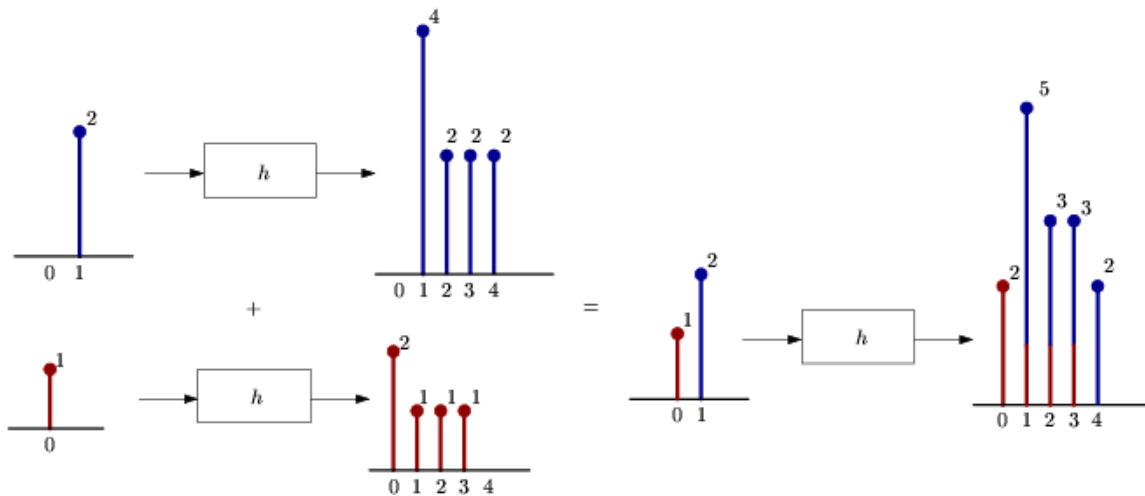


Figure 4.7: We now use the additivity portion of the linearity property of the system to complete the picture. Since any discrete-time signal is just a sum of scaled and shifted discrete-time impulses, we can find the output from knowing the input and the impulse response.

4.2.4 Convolution Through Time (A Graphical Approach)

In this section we will develop a second graphical interpretation of discrete-time convolution. We will begin this by writing the convolution sum allowing x to be a causal, length- m signal and h to be a causal, length- k ,

LTI system. This gives us the finite summation,

$$y[n] = \sum_{l=0}^{m-1} (x[l] h[n-l]) \quad (4.12)$$

Notice that for any given n we have a sum of the products of x_l and a time-delayed h_{-l} . This is to say that we multiply the terms of x by the terms of a time-reversed h and add them up.

Going back to the previous example:

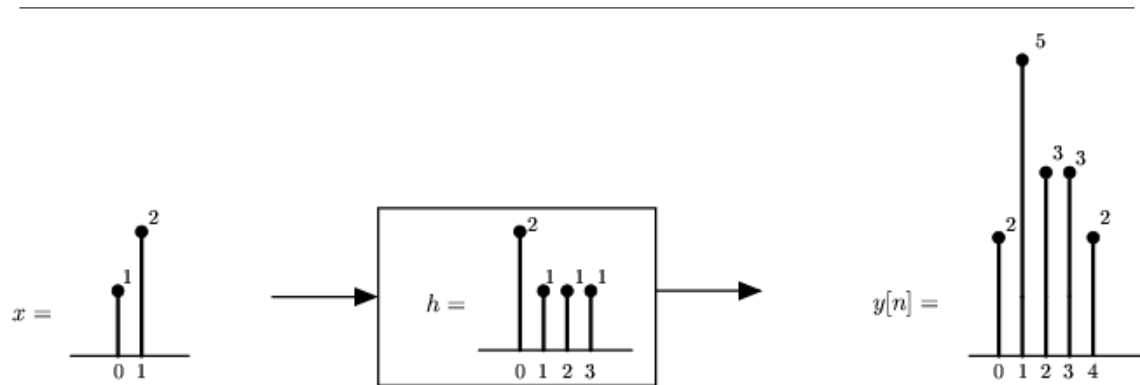


Figure 4.8: This is the end result that we are looking to find.

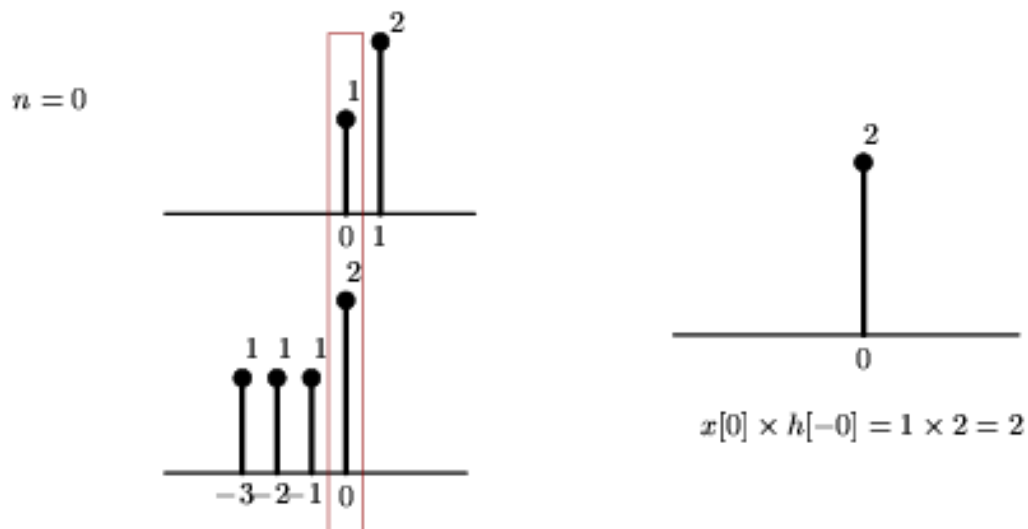


Figure 4.9: Here we reverse the impulse response, h , and begin its traverse at time 0.

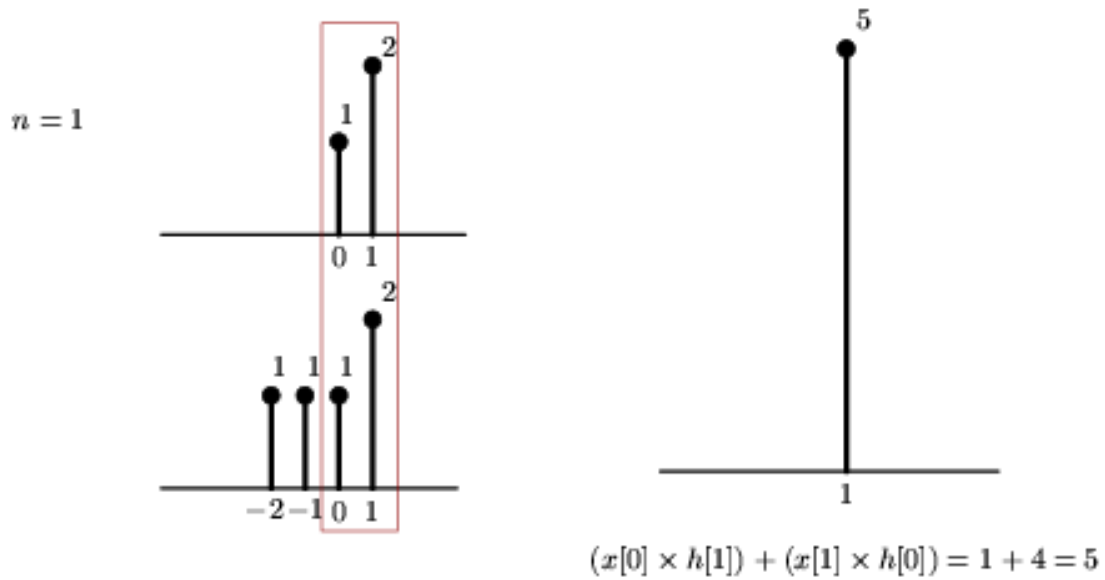


Figure 4.10: We continue the traverse. See that at time 1, we are multiplying two elements of the input signal by two elements of the impulse response.

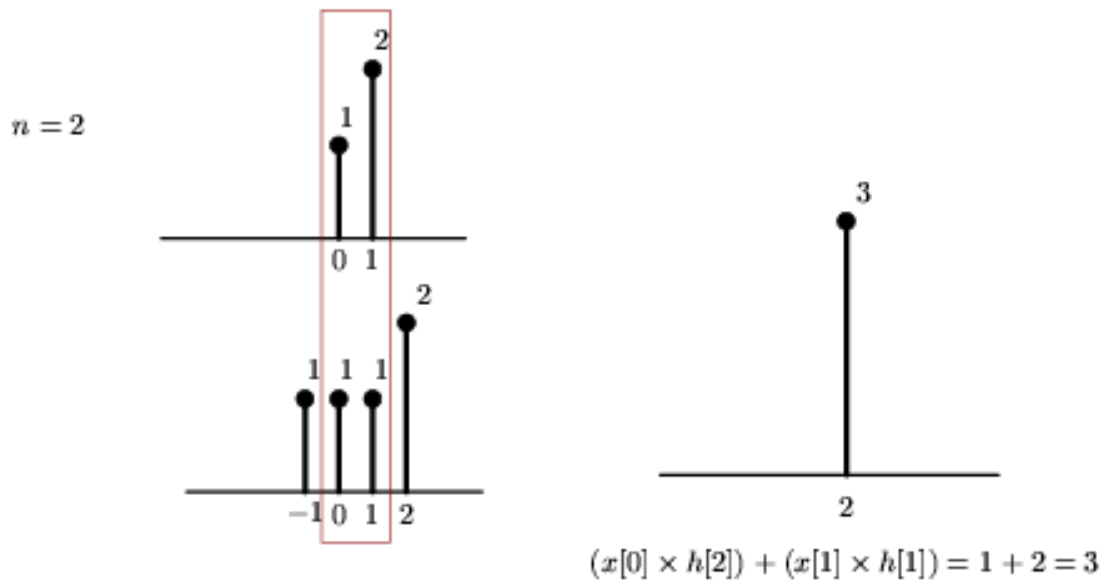


Figure 4.11

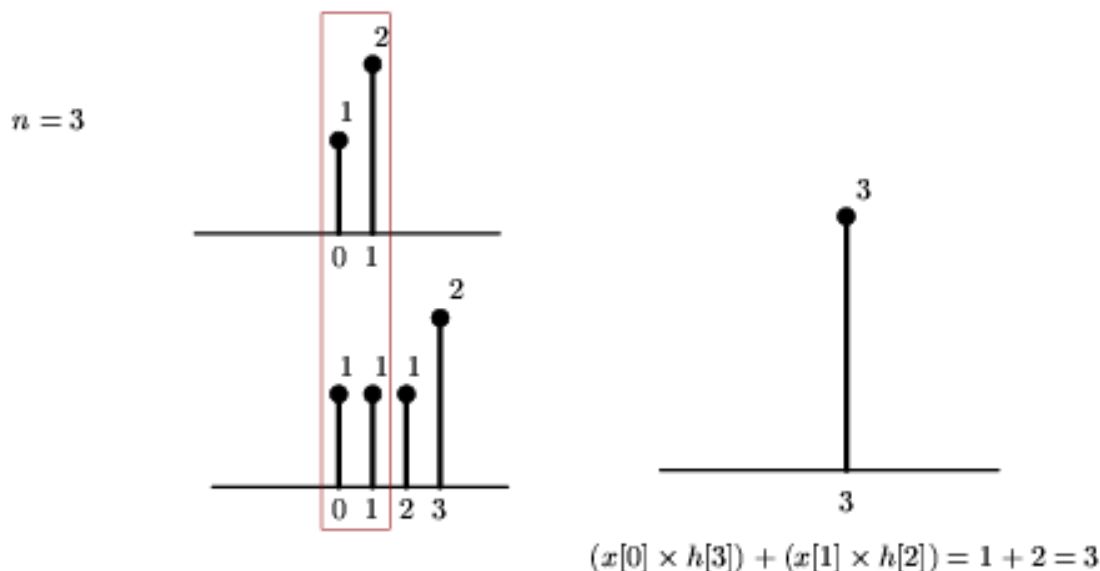


Figure 4.12: If we follow this through to one more step, $n = 4$, then we can see that we produce the same output as we saw in the initial example.

What we are doing in the above demonstration is reversing the impulse response in time and "walking it across" the input signal. Clearly, this yields the same result as scaling, shifting and summing impulse responses.

This approach of time-reversing, and sliding across is a common approach to presenting convolution, since it demonstrates how convolution builds up an output through time.

4.3 Circular Convolution and the DFT⁸

4.3.1 Introduction

You should be familiar with Discrete-Time Convolution (Section 4.2), which tells us that given two discrete-time signals $x[n]$, the system's input, and $h[n]$, the system's response, we define the output of the system as

$$\begin{aligned} y[n] &= x[n] * h[n] \\ &= \sum_{k=-\infty}^{\infty} (x[k] h[n-k]) \end{aligned} \quad (4.13)$$

When we are given two DFTs (finite-length sequences usually of length N), we cannot just multiply them together as we do in the above convolution formula, often referred to as **linear convolution**. Because the DFTs are periodic, they have nonzero values for $n \geq N$ and thus the multiplication of these two DFTs will be nonzero for $n \geq N$. We need to define a new type of convolution operation that will result in our convolved signal being zero outside of the range $n = \{0, 1, \dots, N-1\}$. This idea led to the development of **circular convolution**, also called cyclic or periodic convolution.

⁸This content is available online at <http://cnx.org/content/m10786/2.8/>.

4.3.2 Circular Convolution Formula

What happens when we multiply two DFT's together, where $Y[k]$ is the DFT of $y[n]$?

$$Y[k] = F[k] H[k] \quad (4.14)$$

when $0 \leq k \leq N-1$

Using the DFT synthesis formula for $y[n]$

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} \left(F[k] H[k] e^{j \frac{2\pi}{N} kn} \right) \quad (4.15)$$

And then applying the analysis formula $F[k] = \sum_{m=0}^{N-1} \left(f[m] e^{(-j) \frac{2\pi}{N} km} \right)$

$$\begin{aligned} y[n] &= \frac{1}{N} \sum_{k=0}^{N-1} \left(\left(\sum_{m=0}^{N-1} \left(f[m] e^{(-j) \frac{2\pi}{N} km} \right) \right) H[k] e^{j \frac{2\pi}{N} kn} \right) \\ &= \sum_{m=0}^{N-1} \left(f[m] \left(\frac{1}{N} \sum_{k=0}^{N-1} \left(H[k] e^{j \frac{2\pi}{N} k(n-m)} \right) \right) \right) \end{aligned} \quad (4.16)$$

where we can reduce the second summation found in the above equation into $h[((n-m))_N] = \frac{1}{N} \sum_{k=0}^{N-1} \left(H[k] e^{j \frac{2\pi}{N} k(n-m)} \right)$

$$y[n] = \sum_{m=0}^{N-1} \left(f[m] h[((n-m))_N] \right)$$

which equals circular convolution! When we have $0 \leq n \leq N-1$ in the above, then we get:

$$y[n] \equiv (f[n] \circledast h[n]) \quad (4.17)$$

NOTE: The notation \circledast represents cyclic convolution "mod N".

4.3.2.1 Steps for Cyclic Convolution

Steps for cyclic convolution are the same as the usual convo, except all index calculations are done "mod N" = "on the wheel"

Steps for Cyclic Convolution

- Step 1: "Plot" $f[m]$ and $h[(-m)_N]$

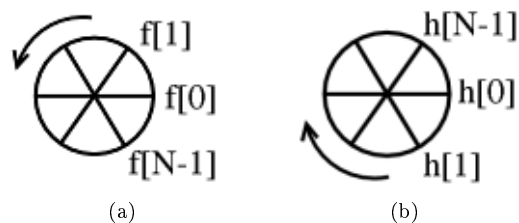


Figure 4.13: Step 1

- Step 2: "Spin" $h[(-m)_N]$ n notches ACW (counter-clockwise) to get $h[(n-m)_N]$ (i.e. Simply rotate the sequence, $h[n]$, clockwise by n steps).

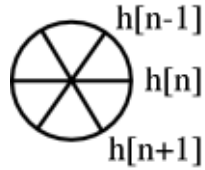


Figure 4.14: Step 2

- Step 3: Pointwise multiply the $f[m]$ wheel and the $h[(n-m)_N]$ wheel.

$$sum = y[n]$$

- Step 4: Repeat for all $0 \leq n \leq N - 1$

Example 4.3: Convolve ($n = 4$)

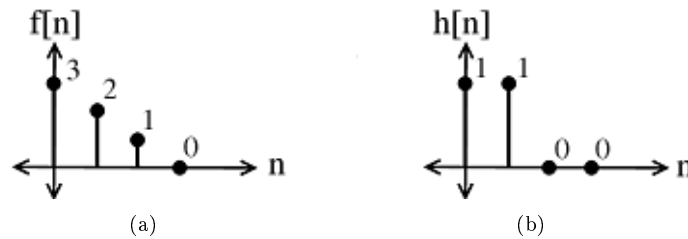


Figure 4.15: Two discrete-time signals to be convolved.

- $h[(-m)_N]$

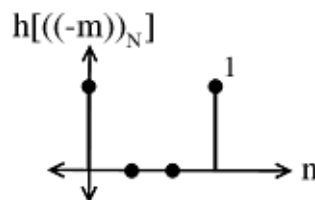


Figure 4.16

Multiply $f[m]$ and sum to yield: $y[0] = 3$

- $h[(1-m)_N]$

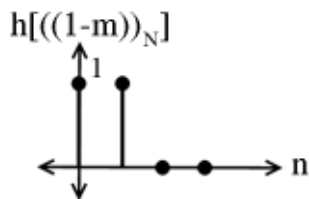


Figure 4.17

Multiply $f[m]$ and sum to yield: $y[1] = 5$

- $h[(2-m)_N]$

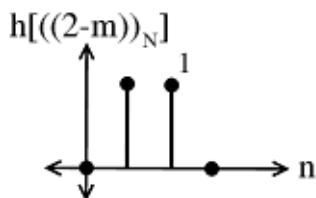


Figure 4.18

Multiply $f[m]$ and sum to yield: $y[2] = 3$

- $h[(3-m)_N]$

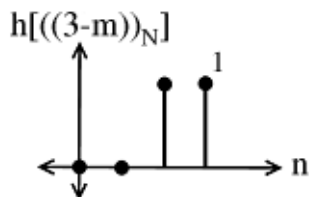


Figure 4.19

Multiply $f[m]$ and sum to yield: $y[3] = 1$

Example 4.4

The following demonstration allows you to explore this algorithm for circular convolution. See [here](#)⁹ for instructions on how to use the demo.

⁹"How to use the LabVIEW demos" <<http://cnx.org/content/m11550/latest/>>

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10786/latest/DTCircularConvolution.llb>

4.3.2.2 Alternative Algorithm

Alternative Circular Convolution Algorithm

- Step 1: Calculate the DFT of $f[n]$ which yields $F[k]$ and calculate the DFT of $h[n]$ which yields $H[k]$.
- Step 2: Pointwise multiply $Y[k] = F[k]H[k]$
- Step 3: Inverse DFT $Y[k]$ which yields $y[n]$

Seems like a roundabout way of doing things, *but* it turns out that there are *extremely* fast ways to calculate the DFT of a sequence.

To circularly convolve 2 N -point sequences:

$$y[n] = \sum_{m=0}^{N-1} (f[m] h[(n-m)_N])$$

For each n : N multiples, $N-1$ additions

N points implies N^2 multiplications, $N(N-1)$ additions implies $O(N^2)$ complexity.

4.4 Linear Constant-Coefficient Difference Equations¹⁰

- remember linear differential equations?

$$\frac{d}{dt}y(t) - y(t) = x(t)$$

- A **difference equation** is the discrete-time analogue of a *differential equation*. We simply use differences ($x[n] - x[n-1]$) rather than derivatives ($\frac{d}{dt}x(t)$).
- An important *subclass* of linear systems consists of those whose input $x[n]$ and output $y[n]$ obey an N -th order LCCDE:

$$\sum_{K=0}^N (a_K y[n-K]) = \sum_{K=0}^M (b_K x[n-K]) \quad (4.18)$$

Example 4.5: Moving average system

$$y[n] = \frac{1}{M_1 + M_2 + 1} \sum_{K=M_1}^{M_2} (x[n-K])$$

where set $M_1 = 0$ and $M_2 = M$, $a_K = \begin{cases} 1 & \text{if } K = 0 \\ 0 & \text{otherwise} \end{cases}$, $b_K = \begin{cases} \frac{1}{M+1} & \text{if } 0 \leq K \leq M \\ 0 & \text{otherwise} \end{cases}$

¹⁰This content is available online at <<http://cnx.org/content/m12325/1.3/>>.

4.4.1 How to Implement?

code / hardware $M = 2$: $y[n] = \frac{1}{3} \sum_{K=0}^2 (x[n - K])$
 IIR? FIR?

Example 4.6: Recursive System

$$y[n] = \sum_{K=1}^N (\alpha_K y[n - K]) + x[n]$$

$$\sum_{K=0}^N (a_K y[n - K]) = x[n]$$

$$\text{Where } a_K = \begin{cases} 1 & \text{if } K = 0 \\ -\alpha_K & \text{if } 1 \leq K \leq N \\ 0 & \text{otherwise} \end{cases}$$

4.4.1 How to Implement?

$N = 2$: $y[n] = \sum_{K=1}^2 (\alpha_K y[n - K]) + x[n]$
 IIR? FIR?

FIR ~ MOVING AVERAGE ~ FEED FORWARD

IIR ~ RECURSIVE ~ FEED BACK

- The *SOLUTION* of a difference equation is similar to a differential equation.
- In particular, note that a single input-output pair ($x[n]$, $y_p[n]$) that solves the DE is not enough to characterize the solution.

$$\sum_{k=0}^N (a_k y_p[n - k]) = \sum_{k=0}^M (b_k x[n - k])$$

Add in zero to get the homogenous equation: $\sum_{k=0}^N (a_k y_h[n - k]) = 0$.

$$\sum_{k=0}^N (a_k (y_p[n - k] + y_h[n - k])) = \sum_{k=0}^M (b_k x[n - k])$$

where the particular solution ("forced") $y_p[n]$ and homogeneous solution ("unforced") $y_h[n]$

General Solution

$$y(n) = y_p(n) + y_h(n) \quad (4.19)$$

4.5 Solving Linear Constant-Coefficient Difference Equations¹¹

- **Step 1** - Given the input $x[n]$, find a solution to

$$\sum_{k=0}^N (a_k y_p[n - k]) = \left(\sum_{k=0}^M (b_k x[n - k]) \right)$$

NOTE: Just any old solution will do!

$y_p[n]$ - particular solution.

¹¹This content is available online at <<http://cnx.org/content/m12326/1.3/>>.

- Solve the homogeneous equation

$$\sum_{k=0}^N (a_k y_h[n-k]) = 0$$

for $y_h[n]$ - homogeneous solution.

- Complete solution given by

$$y[n] = y_p[n] + y_h[n]$$

4.5.1 Solving The Homogeneous Equation

- What does it mean?



Figure 4.20

- Clearly $y_h[n]$ depends on the *INITIAL CONDITIONS* of the system T .

- Linearity
- Time-Invariance
- Causality

will each depend on these conditions.

- In this course, we will emphasize the simplest case, when T is "initially at rest" with "zero initial conditions." → we will get LTI and causal solutions. (although possibly at the expense of *stability*)

Example 4.7

Solve

$$y[n] - ay[n-1] = x[n]$$

where $|a| < 1$ for $x[n] = \delta[n]$.

- **Step 1: Particular Solution** - Assume $n \geq 0$ and "zero initial conditions"

$$y_p[0] = (\delta[0] \rightarrow 1) + a(y_p[-1] \rightarrow 0) = 1$$

$$y_p[1] = (\delta[1] \rightarrow 0) + a(y_p[0] \rightarrow 1) = a$$

$$y_p[2] = (\delta[2] \rightarrow 0) + a(y_p[1] \rightarrow a) = a^2$$

$$y_p[n] = a^n \tag{4.20}$$

where $n \geq 0$

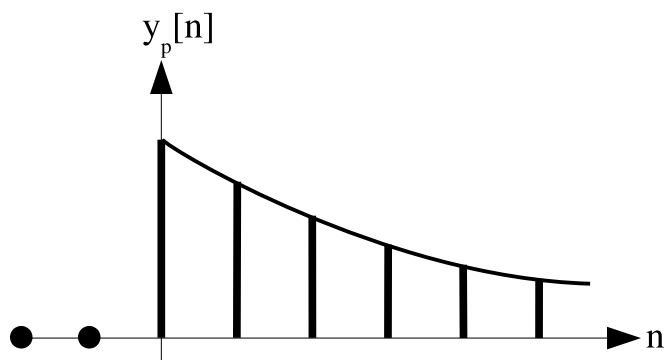


Figure 4.21

- **Step 2: Homogeneous Solution** - If $x[n] = 0$, then $y_h[n] - ay_h[n-1] = 0$

$$y_h[n] = ay_h[n-1]$$

A solution is given by

$$y_h[n] = ca^n \tag{4.21}$$

for all n .

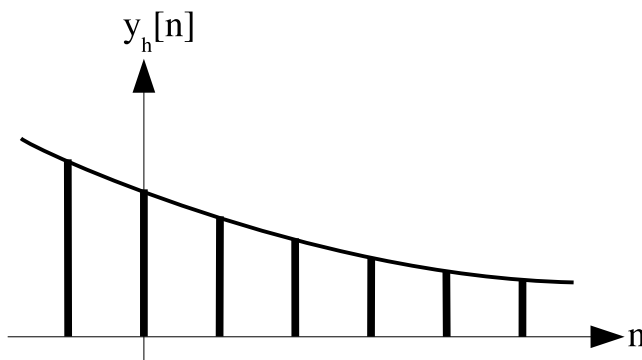


Figure 4.22

- **Step 3: Reconcile** -

$$y[n] = y_n[p] + y_h[n]$$

$$= a^n u[n] + ca^n$$

How to pick c ? Need auxiliary conditions.

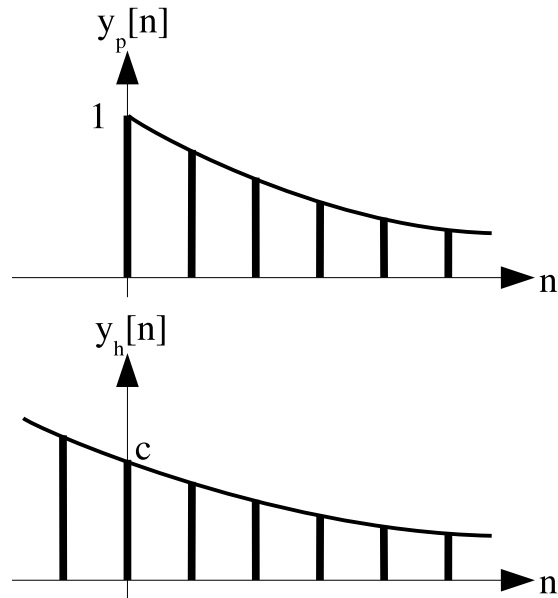


Figure 4.23

If we desire a *causal system*, then $c = 0$ and

$$y[n] = a^n u[n]$$

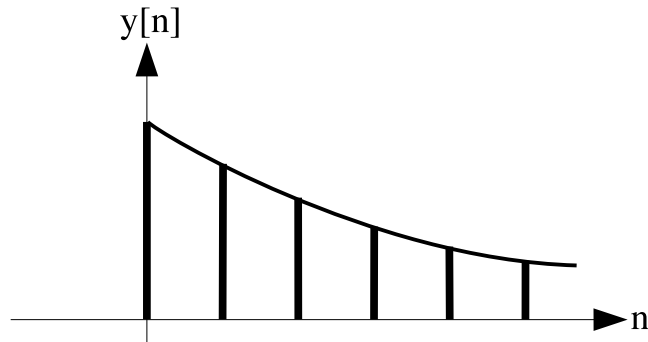


Figure 4.24

If we desire an *anticausal system* then choose $c = -1$, so

$$y[n] = -(a^n u[-n])$$

This does not assume "system initially at rest!"

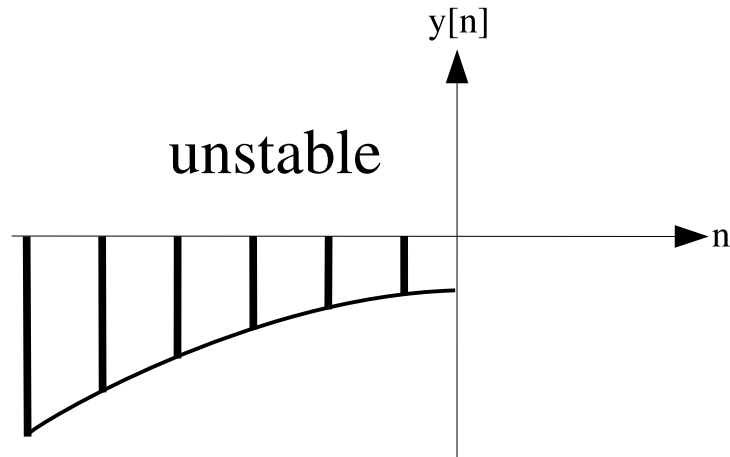


Figure 4.25

Notes

1. Solution 1 (p. 86) was causal and stable.
2. Solution 2 (p. 86) was anticausal and unstable.

In general, linearity, time-invariance, and causality of a system implemented as a DE will depend on the auxiliary conditions.

FACT: If we assume that the system is initially at **rest** ("zero initial conditions"), then it will be **LINEAR, TIME-INVARIANT, and CAUSAL**.

Note:

Setting input $\mathbf{x} = \delta_0$ impulse and setting initial conditions all = 0 and solving for y_p yields

$$y_p = \mathbf{h}$$

as the impulse response of this LSI system.

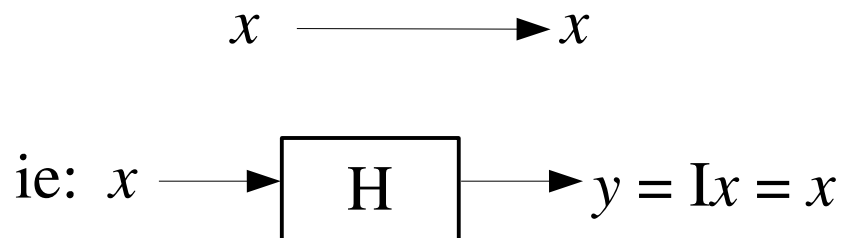
Example 4.8: Frequency Response of a "wire"

Figure 4.26

Impulse Response:

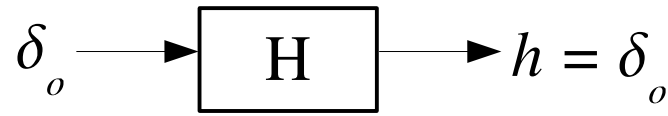


Figure 4.27

so Frequency Response:

$$H = F^H \delta_0 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} (h[n] e^{-j\frac{2\pi}{N}kn}) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} (\delta_0[n] e^{-j\frac{2\pi}{N}kn})$$

$$\delta_0[n] = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$= \frac{1}{\sqrt{N}}$$

Flat

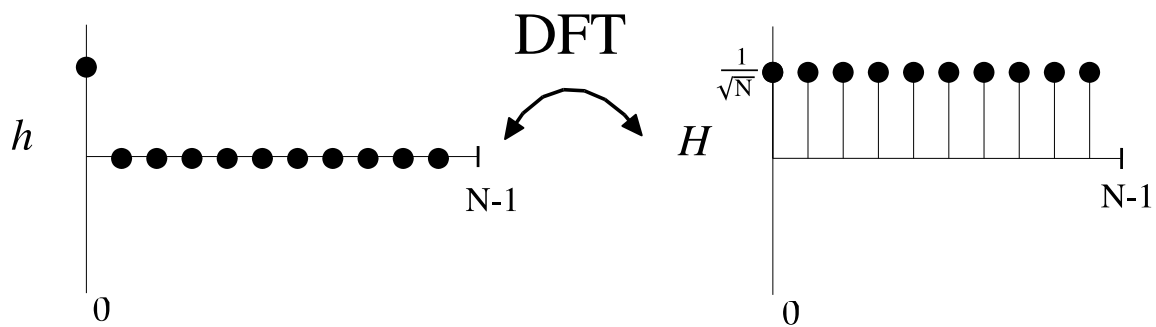


Figure 4.28

Solutions to Exercises in Chapter 4

Solution to Exercise 4.1 (p. 70)

The indices can be negative, and this condition is not allowed in MATLAB. To fix it, we must start the signals later in the array.

Solution to Exercise 4.2 (p. 72)

Such terms would require the system to know what future input or output values would be before the current value was computed. Thus, such terms can cause difficulties.

Chapter 5

Linear Algebra Overview

5.1 Linear Algebra: The Basics¹

This brief tutorial on some key terms in linear algebra is not meant to replace or be very helpful to those of you trying to gain a deep insight into linear algebra. Rather, this brief introduction to some of the terms and ideas of linear algebra is meant to provide a little background to those trying to get a better understanding or learn about eigenvectors and eigenfunctions, which play a big role in deriving a few important ideas on Signals and Systems. The goal of these concepts will be to provide a background for signal decomposition and to lead up to the derivation of the Fourier Series (Section 6.2).

5.1.1 Linear Independence

A set of vectors $\{x_1, x_2, \dots, x_k\}$, $x_i \in \mathbb{C}^n$ are **linearly independent** if none of them can be written as a linear combination of the others.

Definition 5: Linearly Independent

For a given set of vectors, $\{x_1, x_2, \dots, x_n\}$, they are linearly independent if

$$c_1x_1 + c_2x_2 + \dots + c_nx_n = 0$$

only when $c_1 = c_2 = \dots = c_n = 0$

Example

We are given the following two vectors:

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$x_2 = \begin{pmatrix} -6 \\ -4 \end{pmatrix}$$

These are *not linearly independent* as proven by the following statement, which, by inspection, can be seen to not adhere to the definition of linear independence stated above.

$$x_2 = -2x_1 \Rightarrow 2x_1 + x_2 = 0$$

Another approach to reveal a vectors independence is by graphing the vectors. Looking at these two vectors geometrically (as in Figure 5.1), one can again prove that these vectors are *not* linearly independent.

¹This content is available online at <<http://cnx.org/content/m10734/2.4/>>.

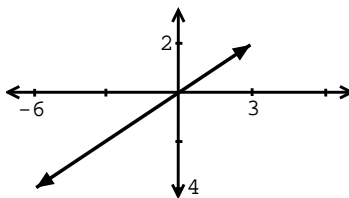


Figure 5.1: Graphical representation of two vectors that are not linearly independent.

Example 5.1

We are given the following two vectors:

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$x_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

These are *linearly independent* since

$$c_1 x_1 = -(c_2 x_2)$$

only if $c_1 = c_2 = 0$. Based on the definition, this proof shows that these vectors are indeed linearly independent. Again, we could also graph these two vectors (see Figure 5.2) to check for linear independence.

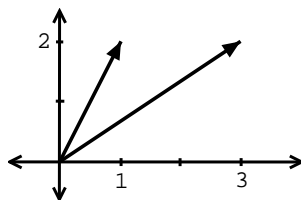


Figure 5.2: Graphical representation of two vectors that are linearly independent.

Exercise 5.1

Are $\{x_1, x_2, x_3\}$ linearly independent?

(Solution on p. 109.)

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$x_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$x_3 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

As we have seen in the two above examples, often times the independence of vectors can be easily seen through a graph. However this may not be as easy when we are given three or more vectors. Can you easily tell whether or not these vectors are independent from Figure 5.3. Probably not, which is why the method used in the above solution becomes important.

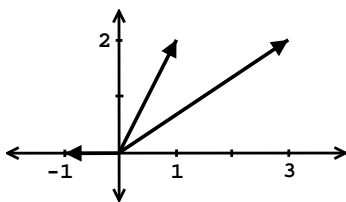


Figure 5.3: Plot of the three vectors. Can be shown that a linear combination exists among the three, and therefore they are *not* linear independent.

HINT: A set of m vectors in \mathbb{C}^n cannot be linearly independent if $m > n$.

5.1.2 Span

Definition 6: Span

The span² of a set of vectors $\{x_1, x_2, \dots, x_k\}$ is the set of vectors that can be written as a linear combination of $\{x_1, x_2, \dots, x_k\}$

$$\text{span}(\{x_1, \dots, x_k\}) = \{\alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_k x_k, \alpha_i \in \mathbb{C}^n\}$$

Example

Given the vector

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

the span of x_1 is a *line*.

Example

Given the vectors

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

²"Subspaces", Definition 2: "Span" <<http://cnx.org/content/m10297/latest/#defn2>>

$$x_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

the span of these vectors is \mathbb{C}^2 .

5.1.3 Basis

Definition 7: Basis

A basis for \mathbb{C}^n is a set of vectors that: (1) spans \mathbb{C}^n and (2) is linearly independent. Clearly, any set of n linearly independent vectors is a **basis** for \mathbb{C}^n .

Example 5.2

We are given the following vector

$$e_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where the 1 is always in the i th place and the remaining values are zero. Then the **basis** for \mathbb{C}^n is

$$\{e_i \ , \ i = [1, 2, \dots, n] \}$$

NOTE: $\{e_i \ , \ i = [1, 2, \dots, n] \}$ is called the **standard basis**.

Example 5.3

$$h_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$h_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$\{h_1, h_2\}$ is a basis for \mathbb{C}^2 .

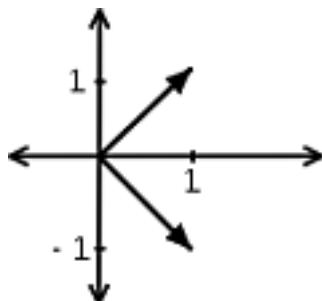


Figure 5.4: Plot of basis for \mathbb{C}^2

If $\{b_1, \dots, b_n\}$ is a basis for \mathbb{C}^n , then we can express *any* $\mathbf{x} \in \mathbb{C}^n$ as a linear combination of the b_i 's:

$$\mathbf{x} = \alpha_1 b_1 + \alpha_2 b_2 + \dots + \alpha_n b_n, \quad \alpha_i \in \mathbb{C}$$

Example 5.4

Given the following vector,

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

writing \mathbf{x} in terms of $\{e_1, e_2\}$ gives us

$$\mathbf{x} = e_1 + 2e_2$$

Exercise 5.2

(Solution on p. 109.)

Try and write \mathbf{x} in terms of $\{h_1, h_2\}$ (defined in the previous example).

In the two basis examples above, \mathbf{x} is the same vector in both cases, but we can express it in many different ways (we give only two out of many, many possibilities). You can take this even further by extending this idea of a basis to **function spaces**.

NOTE: As mentioned in the introduction, these concepts of linear algebra will help prepare you to understand the Fourier Series (Section 6.2), which tells us that we can express periodic functions, $f(t)$, in terms of their basis functions, $e^{j\omega_0 nt}$.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10734/latest/LinearAlgebraCalc3.llb>

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10734/latest/LinearTransform.llb>

5.2 Eigenvectors and Eigenvalues³

In this section, our linear systems will be $n \times n$ matrices of complex numbers. For a little background into some of the concepts that this module is based on, refer to the basics of linear algebra (Section 5.1).

5.2.1 Eigenvectors and Eigenvalues

Let A be an $n \times n$ matrix, where A is a linear operator on vectors in \mathbb{C}^n .

$$A\mathbf{x} = \mathbf{b} \quad (5.1)$$

where \mathbf{x} and \mathbf{b} are $n \times 1$ vectors (Figure 5.5).

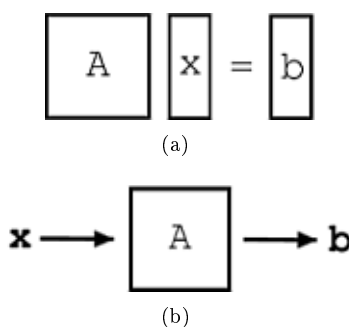


Figure 5.5: Illustration of linear system and vectors.

Definition 8: eigenvector

An eigenvector of A is a vector $\mathbf{v} \in \mathbb{C}^n$ such that

$$A\mathbf{v} = \lambda\mathbf{v} \quad (5.2)$$

where λ is called the corresponding **eigenvalue**. A only changes the *length* of \mathbf{v} , not its direction.

5.2.1.1 Graphical Model

Through Figure 5.6 and Figure 5.7, let us look at the difference between (5.1) and (5.2).

³This content is available online at <http://cnx.org/content/m10736/2.7/>.

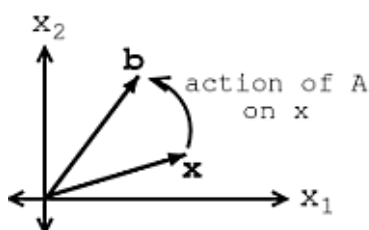


Figure 5.6: Represents (5.1), $Ax = b$.

If \mathbf{v} is an eigenvector of A , then only its length changes. See Figure 5.7 and notice how our vector's length is simply scaled by our variable, λ , called the **eigenvalue**:

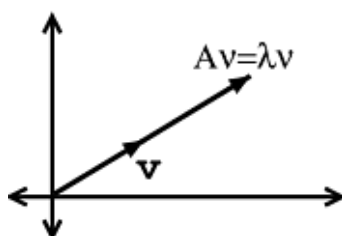


Figure 5.7: Represents (5.2), $A\mathbf{v} = \lambda\mathbf{v}$.

NOTE: When dealing with a matrix A , eigenvectors are the *simplest* possible vectors to operate on.

5.2.1.2 Examples

Exercise 5.3

(Solution on p. 109.)

From inspection and understanding of eigenvectors, find the two eigenvectors, v_1 and v_2 , of

$$A = \begin{pmatrix} 3 & 0 \\ 0 & -1 \end{pmatrix}$$

Also, what are the corresponding eigenvalues, λ_1 and λ_2 ? Do not worry if you are having problems seeing these values from the information given so far, we will look at more rigorous ways to find these values soon.

Exercise 5.4

(Solution on p. 109.)

Show that these two vectors,

$$v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$v_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

are eigenvectors of A , where $A = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$. Also, find the corresponding eigenvalues.

5.2.2 Calculating Eigenvalues and Eigenvectors

In the above examples, we relied on your understanding of the definition and on some basic observations to find and prove the values of the eigenvectors and eigenvalues. However, as you can probably tell, finding these values will not always be that easy. Below, we walk through a rigorous and mathematical approach at calculating the eigenvalues and eigenvectors of a matrix.

5.2.2.1 Finding Eigenvalues

Find $\lambda \in \mathbb{C}$ such that $\mathbf{v} \neq \mathbf{0}$, where $\mathbf{0}$ is the "zero vector." We will start with (5.2), and then work our way down until we find a way to explicitly calculate λ .

$$A\mathbf{v} = \lambda\mathbf{v}$$

$$A\mathbf{v} - \lambda\mathbf{v} = 0$$

$$(A - \lambda I)\mathbf{v} = 0$$

In the previous step, we used the fact that

$$\lambda\mathbf{v} = \lambda I\mathbf{v}$$

where I is the identity matrix.

$$I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & \dots & \dots & 1 \end{pmatrix}$$

So, $A - \lambda I$ is just a new matrix.

Example 5.5

Given the following matrix, A , then we can find our new matrix, $A - \lambda I$.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

$$A - \lambda I = \begin{pmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{pmatrix}$$

If $(A - \lambda I)\mathbf{v} = 0$ for some $\mathbf{v} \neq 0$, then $A - \lambda I$ is *not invertible*. This means:

$$\det(A - \lambda I) = 0$$

This determinant (shown directly above) turns out to be a polynomial expression (of order n). Look at the examples below to see what this means.

Example 5.6

Starting with matrix A (shown below), we will find the polynomial expression, where our eigenvalues will be the dependent variable.

$$A = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$$

$$A - \lambda I = \begin{pmatrix} 3 - \lambda & -1 \\ -1 & 3 - \lambda \end{pmatrix}$$

$$\det(A - \lambda I) = (3 - \lambda)^2 - (-1)^2 = \lambda^2 - 6\lambda + 8$$

$$\lambda = \{2, 4\}$$

Example 5.7

Starting with matrix A (shown below), we will find the polynomial expression, where our eigenvalues will be the dependent variable.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

$$A - \lambda I = \begin{pmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{pmatrix}$$

$$\det(A - \lambda I) = \lambda^2 - (a_{11} + a_{22})\lambda - a_{21}a_{12} + a_{11}a_{22}$$

If you have not already noticed it, calculating the eigenvalues is equivalent to calculating the roots of

$$\det(A - \lambda I) = c_n\lambda^n + c_{n-1}\lambda^{n-1} + \dots + c_1\lambda + c_0 = 0$$

CONCLUSION: Therefore, by simply using calculus to solve for the roots of our polynomial we can easily find the eigenvalues of our matrix.

5.2.2.2 Finding Eigenvectors

Given an eigenvalue, λ_i , the associated eigenvectors are given by

$$A\mathbf{v} = \lambda_i\mathbf{v}$$

$$A \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \lambda_1 v_1 \\ \vdots \\ \lambda_n v_n \end{pmatrix}$$

set of n equations with n unknowns. Simply *solve the n equations* to find the eigenvectors.

5.2.3 Main Point

Say the eigenvectors of A , $\{v_1, v_2, \dots, v_n\}$, span (Section 5.1.2: Span) \mathbb{C}^n , meaning $\{v_1, v_2, \dots, v_n\}$ are linearly independent (Section 5.1.1: Linear Independence) and we can write any $\mathbf{x} \in \mathbb{C}^n$ as

$$\mathbf{x} = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n \quad (5.3)$$

where $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \in \mathbb{C}$. All that we are doing is rewriting \mathbf{x} in terms of eigenvectors of A . Then,

$$A\mathbf{x} = A(\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n)$$

$$A\mathbf{x} = \alpha_1 A v_1 + \alpha_2 A v_2 + \dots + \alpha_n A v_n$$

$$A\mathbf{x} = \alpha_1 \lambda_1 v_1 + \alpha_2 \lambda_2 v_2 + \dots + \alpha_n \lambda_n v_n = b$$

Therefore we can write,

$$\mathbf{x} = \sum_i (\alpha_i v_i)$$

and this leads us to the following depicted system:

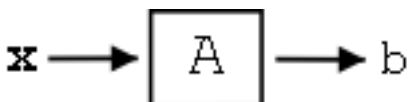


Figure 5.8: Depiction of system where we break our vector, \mathbf{x} , into a sum of its eigenvectors.

where in Figure 5.8 we have,

$$b = \sum_i (\alpha_i \lambda_i v_i)$$

MAIN POINT: By breaking up a vector, \mathbf{x} , into a combination of eigenvectors, the calculation of $A\mathbf{x}$ is broken into "easy to swallow" pieces.

5.2.4 Practice Problem

Exercise 5.5

(Solution on p. 109.)

For the following matrix, A and vector, \mathbf{x} , solve for their product. Try solving it using two different methods: directly and using eigenvectors.

$$A = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

5.3 Matrix Diagonalization⁴

From our understanding of eigenvalues and eigenvectors (Section 5.2) we have discovered several things about our operator matrix, A . We know that if the eigenvectors of A span \mathbb{C}^n and we know how to express any vector \mathbf{x} in terms of $\{v_1, v_2, \dots, v_n\}$, then we have the operator A all figured out. If we have A acting on \mathbf{x} , then this is equal to A acting on the combinations of eigenvectors. Which we know proves to be fairly easy!

We are still left with two questions that need to be addressed:

1. When do the eigenvectors $\{v_1, v_2, \dots, v_n\}$ of A span \mathbb{C}^n (assuming $\{v_1, v_2, \dots, v_n\}$ are linearly independent)?
2. How do we express a given vector \mathbf{x} in terms of $\{v_1, v_2, \dots, v_n\}$?

5.3.1 Answer to Question #1

QUESTION 1: When do the eigenvectors $\{v_1, v_2, \dots, v_n\}$ of A span \mathbb{C}^n ?

If A has n distinct eigenvalues

$$\lambda_i \neq \lambda_j \quad , \quad i \neq j$$

where i and j are integers, then A has n linearly independent eigenvectors $\{v_1, v_2, \dots, v_n\}$ which then span \mathbb{C}^n .

ASIDE: The proof of this statement is not very hard, but is not really interesting enough to include here. If you wish to research this idea further, read Strang, G., "Linear Algebra and its Application" for the proof.

Furthermore, n distinct eigenvalues means

$$\det(A - \lambda I) = c_n \lambda^n + c_{n-1} \lambda^{n-1} + \dots + c_1 \lambda + c_0 = 0$$

has n distinct roots.

5.3.2 Answer to Question #2

QUESTION 2: How do we express a given vector \mathbf{x} in terms of $\{v_1, v_2, \dots, v_n\}$?

We want to find $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \in \mathbb{C}$ such that

$$\mathbf{x} = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n \tag{5.4}$$

In order to find this set of variables, we will begin by collecting the vectors $\{v_1, v_2, \dots, v_n\}$ as columns in a $n \times n$ matrix V .

$$V = \begin{pmatrix} \vdots & \vdots & & \vdots \\ v_1 & v_2 & \dots & v_n \\ \vdots & \vdots & & \vdots \end{pmatrix}$$

Now (5.4) becomes

$$\mathbf{x} = \begin{pmatrix} \vdots & \vdots & & \vdots \\ v_1 & v_2 & \dots & v_n \\ \vdots & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}$$

⁴This content is available online at <<http://cnx.org/content/m10738/2.5/>>.

or

$$\mathbf{x} = V\alpha$$

which gives us an easy form to solve for our variables in question, α :

$$\alpha = V^{-1}\mathbf{x}$$

Note that V is invertible since it has n linearly independent columns.

5.3.2.1 Aside

Let us recall our knowledge of functions and there basis and examine the role of V .

$$\mathbf{x} = V\alpha$$

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = V \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}$$

where α is just \mathbf{x} expressed in a different basis (Section 5.1.3: Basis):

$$\mathbf{x} = x_1 \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + \cdots + x_n \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

$$\mathbf{x} = \alpha_1 \begin{pmatrix} \vdots \\ v_1 \\ \vdots \end{pmatrix} + \alpha_2 \begin{pmatrix} \vdots \\ v_2 \\ \vdots \end{pmatrix} + \cdots + \alpha_n \begin{pmatrix} \vdots \\ v_n \\ \vdots \end{pmatrix}$$

V transforms \mathbf{x} from the standard basis to the basis $\{v_1, v_2, \dots, v_n\}$

5.3.3 Matrix Diagonalization and Output

We can also use the vectors $\{v_1, v_2, \dots, v_n\}$ to represent the output, \mathbf{b} , of a system:

$$\mathbf{b} = A\mathbf{x} = A(\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n)$$

$$A\mathbf{x} = \alpha_1 \lambda_1 v_1 + \alpha_2 \lambda_2 v_2 + \cdots + \alpha_n \lambda_n v_n = \mathbf{b}$$

$$A\mathbf{x} = \begin{pmatrix} \vdots & \vdots & \vdots \\ v_1 & v_2 & \dots & v_n \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \lambda_1 \alpha_1 \\ \vdots \\ \lambda_n \alpha_n \end{pmatrix}$$

$$A\mathbf{x} = V\Lambda\alpha$$

$$A\mathbf{x} = V\Lambda V^{-1}\mathbf{x}$$

where Λ is the matrix with the eigenvalues down the diagonal:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}$$

Finally, we can cancel out the \mathbf{x} and are left with a final equation for A :

$$A = V\Lambda V^{-1}$$

5.3.3.1 Interpretation

For our interpretation, recall our key formulas:

$$\alpha = V^{-1}\mathbf{x}$$

$$\mathbf{b} = \sum_i (\alpha_i \lambda_i v_i)$$

We can interpret operating on \mathbf{x} with A as:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \rightarrow \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \rightarrow \begin{pmatrix} \lambda_1 \alpha_1 \\ \vdots \\ \lambda_n \alpha_n \end{pmatrix} \rightarrow \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

where the three steps (arrows) in the above illustration represent the following three operations:

1. Transform \mathbf{x} using V^{-1} , which yields α
2. Multiplication by Λ
3. Inverse transform using V , which gives us \mathbf{b}

This is the paradigm we will use for LTI systems!

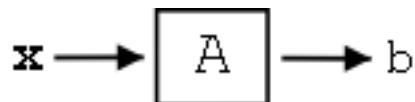


Figure 5.9: Simple illustration of LTI system!

5.4 Eigen-stuff in a Nutshell⁵

5.4.1 A Matrix and its Eigenvector

The reason we are stressing eigenvectors (Section 5.2) and their importance is because the action of a matrix A on one of its eigenvectors \mathbf{v} is

1. extremely easy (and fast) to calculate

$$A\mathbf{v} = \lambda\mathbf{v} \tag{5.5}$$

just *multiply* \mathbf{v} by λ .

2. easy to interpret: A just *scales* \mathbf{v} , keeping its direction constant and only altering the vector's length.

If only every vector were an eigenvector of A ...

5.4.2 Using Eigenvectors' Span

Of course, not every vector can be ... BUT ... For certain matrices (including ones with distinct eigenvalues, λ 's), their eigenvectors span (Section 5.1.2: Span) \mathbb{C}^n , meaning that for *any* $\mathbf{x} \in \mathbb{C}^n$, we can find $\{\alpha_1, \alpha_2, \alpha_n\} \in \mathbb{C}$ such that:

$$\mathbf{x} = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n \tag{5.6}$$

Given (5.6), we can rewrite $A\mathbf{x} = \mathbf{b}$. This equation is modeled in our LTI system pictured below:

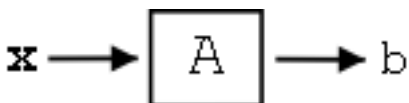


Figure 5.10: LTI System.

$$\mathbf{x} = \sum_i (\alpha_i v_i)$$

$$\mathbf{b} = \sum_i (\alpha_i \lambda_i v_i)$$

The LTI system above represents our (5.5). Below is an illustration of the steps taken to go from \mathbf{x} to \mathbf{b} .

$$\mathbf{x} \rightarrow (\alpha = V^{-1}\mathbf{x}) \rightarrow (\Lambda V^{-1}\mathbf{x}) \rightarrow V\Lambda V^{-1}\mathbf{x} = \mathbf{b}$$

where the three steps (arrows) in the above illustration represent the following three operations:

1. Transform \mathbf{x} using V^{-1} - yields α
2. Action of A in new basis - a multiplication by Λ
3. Translate back to old basis - inverse transform using a multiplication by V , which gives us \mathbf{b}

⁵This content is available online at <<http://cnx.org/content/m10742/2.4/>>.

5.5 Eigenfunctions of LTI Systems⁶

5.5.1 Introduction

Hopefully you are familiar with the notion of the eigenvectors of a "matrix system," if not they do a quick review of eigen-stuff (Section 5.4). We can develop the same ideas for LTI systems acting on signals. A linear time invariant (LTI) system⁷ \mathcal{H} operating on a continuous input $f(t)$ to produce continuous time output $y(t)$

$$\mathcal{H}[f(t)] = y(t) \quad (5.7)$$

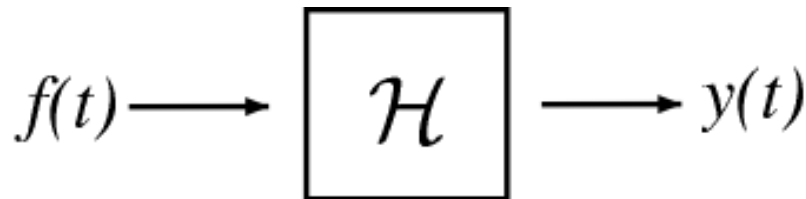


Figure 5.11: $\mathcal{H}[f(t)] = y(t)$. f and t are continuous time (CT) signals and \mathcal{H} is an LTI operator.

is mathematically analogous to an $N \times N$ matrix A operating on a vector $\mathbf{x} \in \mathbb{C}^N$ to produce another vector $\mathbf{b} \in \mathbb{C}^N$ (see Matrices and LTI Systems for an overview).

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (5.8)$$

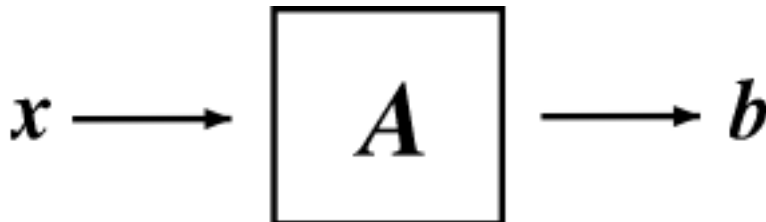


Figure 5.12: $\mathbf{A}\mathbf{x} = \mathbf{b}$ where \mathbf{x} and \mathbf{b} are in \mathbb{C}^N and \mathbf{A} is an $N \times N$ matrix.

Just as an eigenvector (Section 5.2) of \mathbf{A} is a $\mathbf{v} \in \mathbb{C}^N$ such that $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$, $\lambda \in \mathbb{C}$,

⁶This content is available online at <http://cnx.org/content/m10500/2.7/>.

⁷"Introduction to Systems" <http://cnx.org/content/m0005/latest/>

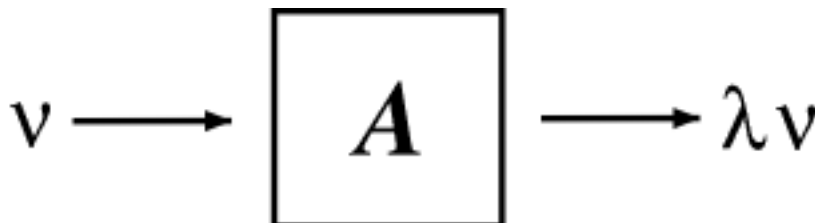


Figure 5.13: $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ where $\mathbf{v} \in \mathbb{C}^N$ is an eigenvector of \mathbf{A} .

we can define an **eigenfunction** (or **eigensignal**) of an LTI system \mathcal{H} to be a signal $f(t)$ such that

$$\mathcal{H}[f(t)] = \lambda f(t) \quad , \quad \lambda \in \mathbb{C} \quad (5.9)$$

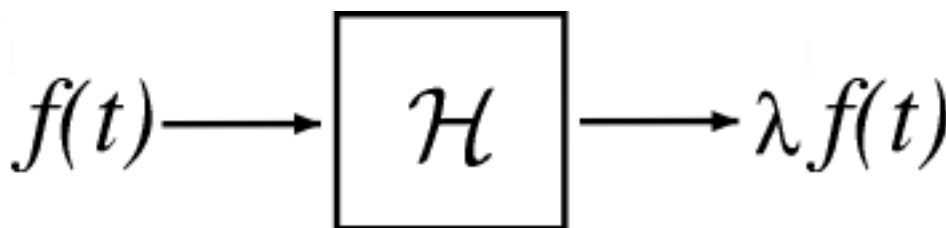


Figure 5.14: $\mathcal{H}[f(t)] = \lambda f(t)$ where f is an eigenfunction of \mathcal{H} .

Eigenfunctions are the *simplest* possible signals for \mathcal{H} to operate on: to calculate the output, we simply multiply the input by a complex number λ .

5.5.2 Eigenfunctions of any LTI System

The class of LTI systems has a set of eigenfunctions in common: the complex exponentials (Section 1.6) e^{st} , $s \in \mathbb{C}$ are eigenfunctions for *all* LTI systems.

$$\mathcal{H}[e^{st}] = \lambda_s e^{st} \quad (5.10)$$

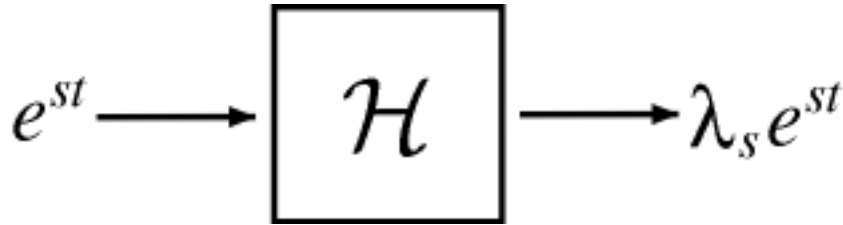


Figure 5.15: $\mathcal{H}[e^{st}] = \lambda_s e^{st}$ where \mathcal{H} is an LTI system.

NOTE: While $\{e^{st}, s \in \mathbb{C}\}$ are always eigenfunctions of an LTI system, they are not necessarily the *only* eigenfunctions.

We can prove (5.10) by expressing the output as a convolution (Section 3.2) of the input e^{st} and the impulse response (Section 1.5) $h(t)$ of \mathcal{H} :

$$\begin{aligned}
 \mathcal{H}[e^{st}] &= \int_{-\infty}^{\infty} h(\tau) e^{s(t-\tau)} d\tau \\
 &= \int_{-\infty}^{\infty} h(\tau) e^{st} e^{-(s\tau)} d\tau \\
 &= e^{st} \int_{-\infty}^{\infty} h(\tau) e^{-(s\tau)} d\tau
 \end{aligned} \tag{5.11}$$

Since the expression on the right hand side does not depend on t , it is a constant, λ_s . Therefore

$$\mathcal{H}[e^{st}] = \lambda_s e^{st} \tag{5.12}$$

The eigenvalue λ_s is a complex number that depends on the exponent s and, of course, the system \mathcal{H} . To make these dependencies explicit, we will use the notation $H(s) \equiv \lambda_s$.

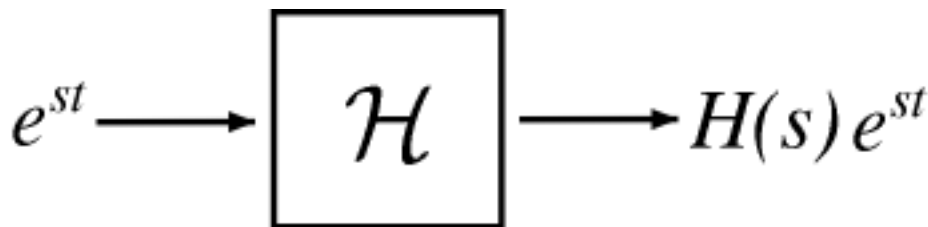


Figure 5.16: e^{st} is the eigenfunction and $H(s)$ are the eigenvalues.

Since the action of an LTI operator on its eigenfunctions e^{st} is easy to calculate and interpret, it is convenient to represent an arbitrary signal $f(t)$ as a linear combination of complex exponentials. The Fourier series (Section 6.2) gives us this representation for periodic continuous time signals, while the (slightly more complicated) Fourier transform⁸ lets us expand arbitrary continuous time signals.

⁸"Derivation of the Fourier Transform" <<http://cnx.org/content/m0046/latest/>>

5.6 Fourier Transform Properties⁹

Short Table of Fourier Transform Pairs

$s(t)$	$S(f)$
$e^{-(at)}u(t)$	$\frac{1}{j2\pi f + a}$
$e^{(-a) t }$	$\frac{2a}{4\pi^2 f^2 + a^2}$
$p(t) = \begin{cases} 1 & \text{if } t < \frac{\Delta}{2} \\ 0 & \text{if } t > \frac{\Delta}{2} \end{cases}$	$\frac{\sin(\pi f \Delta)}{\pi f}$
$\frac{\sin(2\pi W t)}{\pi t}$	$S(f) = \begin{cases} 1 & \text{if } f < W \\ 0 & \text{if } f > W \end{cases}$

Fourier Transform Properties

	Time-Domain	Frequency Domain
Linearity	$a_1 s_1(t) + a_2 s_2(t)$	$a_1 S_1(f) + a_2 S_2(f)$
Conjugate Symmetry	$s(t) \in \mathbb{R}$	$S(f) = S(-f)^*$
Even Symmetry	$s(t) = s(-t)$	$S(f) = S(-f)$
Odd Symmetry	$s(t) = -s(-t)$	$S(f) = -S(-f)$
Scale Change	$s(at)$	$\frac{1}{ a } S\left(\frac{f}{a}\right)$
Time Delay	$s(t - \tau)$	$e^{-(j2\pi f \tau)} S(f)$
Complex Modulation	$e^{j2\pi f_0 t} s(t)$	$S(f - f_0)$
Amplitude Modulation by Cosine	$s(t) \cos(2\pi f_0 t)$	$\frac{S(f - f_0) + S(f + f_0)}{2}$
Amplitude Modulation by Sine	$s(t) \sin(2\pi f_0 t)$	$\frac{S(f - f_0) - S(f + f_0)}{2j}$
Differentiation	$\frac{d}{dt} s(t)$	$j2\pi f S(f)$
Integration	$\int_{-\infty}^t s(\alpha) d\alpha$	$\frac{1}{j2\pi f} S(f)$ if $S(0) = 0$
Multiplication by t	$ts(t)$	$\frac{1}{-(j2\pi)} \frac{d}{df} S(f)$
Area	$\int_{-\infty}^{\infty} s(t) dt$	$S(0)$
Value at Origin	$s(0)$	$\int_{-\infty}^{\infty} S(f) df$
Parseval's Theorem	$\int_{-\infty}^{\infty} s(t) ^2 dt$	$\int_{-\infty}^{\infty} S(f) ^2 df$

⁹This content is available online at <<http://cnx.org/content/m0045/2.8/>>.

Solutions to Exercises in Chapter 5

Solution to Exercise 5.1 (p. 92)

By playing around with the vectors and doing a little trial and error, we will discover the following relationship:

$$x_1 - x_2 + 2x_3 = 0$$

Thus we have found a linear combination of these three vectors that equals zero without setting the coefficients equal to zero. Therefore, these vectors are *not linearly independent!*

Solution to Exercise 5.2 (p. 95)

$$\mathbf{x} = \frac{3}{2}h_1 + \frac{-1}{2}h_2$$

Solution to Exercise 5.3 (p. 97)

The eigenvectors you found should be:

$$v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$v_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

And the corresponding eigenvalues are

$$\lambda_1 = 3$$

$$\lambda_2 = -1$$

Solution to Exercise 5.4 (p. 97)

In order to prove that these two vectors are eigenvectors, we will show that these statements meet the requirements stated in the definition (Definition: "eigenvector", p. 96).

$$Av_1 = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

$$Av_2 = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 4 \\ -4 \end{pmatrix}$$

These results show us that A only scales the two vectors (i.e. changes their length) and thus it proves that (5.2) holds true for the following two eigenvalues that you were asked to find:

$$\lambda_1 = 2$$

$$\lambda_2 = 4$$

If you need more convincing, then one could also easily graph the vectors and their corresponding product with A to see that the results are merely scaled versions of our original vectors, v_1 and v_2 .

Solution to Exercise 5.5 (p. 100)

Direct Method (use basic matrix multiplication)

$$A\mathbf{x} = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 5 \\ 3 \end{pmatrix} = \begin{pmatrix} 12 \\ 4 \end{pmatrix}$$

Eigenvectors (use the eigenvectors and eigenvalues we found earlier for this same matrix)

$$v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$v_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\lambda_1 = 2$$

$$\lambda_2 = 4$$

As shown in (5.3), we want to represent \mathbf{x} as a sum of its scaled eigenvectors. For this case, we have:

$$\mathbf{x} = 4v_1 + v_2$$

$$\mathbf{x} = \begin{pmatrix} 5 \\ 3 \end{pmatrix} = 4 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$A\mathbf{x} = A(4v_1 + v_2) = \lambda_i(4v_1 + v_2)$$

Therefore, we have

$$A\mathbf{x} = 4 \times 2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 4 \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 12 \\ 4 \end{pmatrix}$$

Notice that this method using eigenvectors required *no* matrix multiplication. This may have seemed more complicated here, but just imagine A being really big, or even just a few dimensions larger!

Chapter 6

Continuous Time Fourier Series

6.1 Periodic Signals¹

Recall that a periodic function is a function that repeats itself exactly after some given period, or cycle. We represent the definition of a **periodic function** mathematically as:

$$f(t) = f(t + mT) \quad m \in \mathbb{Z} \quad , \quad (6.1)$$

where $T > 0$ represents the **period**. Because of this, you may also see a signal referred to as a T -periodic signal. Any function that satisfies this equation is periodic.

We can think of periodic functions (with period T) two different ways:

#1) as functions on *all* of \mathbb{R}

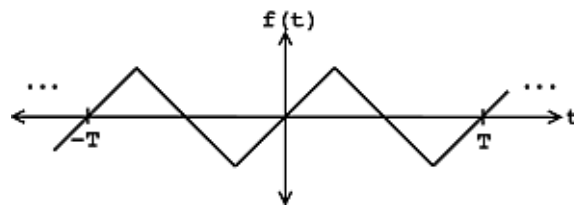


Figure 6.1: Function over all of \mathbb{R} where $f(t_0) = f(t_0 + T)$

#2) or, we can cut out all of the redundancy, and think of them as functions on an interval $[0, T]$ (or, more generally, $[a, a + T]$). If we know the signal is T -periodic then all the information of the signal is captured by the above interval.

¹This content is available online at <http://cnx.org/content/m10744/2.6/>.

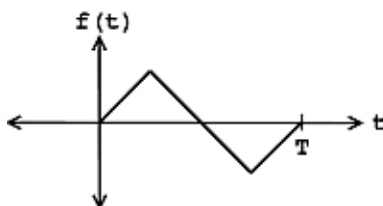


Figure 6.2: Remove the redundancy of the period function so that $f(t)$ is undefined outside $[0, T]$.

An **aperiodic** CT function $f(t)$ does not repeat for any $T \in \mathbb{R}$; i.e. there exists no T s.t. this equation (6.1) holds.

Question: DT definitions?

6.1.1 Continuous-Time

6.1.2 Discrete-Time

Note: Circular vs. Line

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10744/latest/PhaseShift.llb>

6.2 Fourier Series: Eigenfunction Approach²

6.2.1 Introduction

Since complex exponentials (Section 1.6) are eigenfunctions of linear time-invariant (LTI) systems (Section 5.5), calculating the output of an LTI system \mathcal{H} given e^{st} as an input amounts to simple multiplication, where $H(s) \in \mathbb{C}$ is a constant (that depends on s). In the figure (Figure 6.3) below we have a simple exponential input that yields the following output:

$$y(t) = H(s) e^{st} \quad (6.2)$$

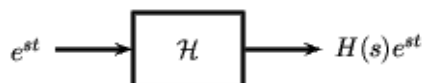


Figure 6.3: Simple LTI system.

²This content is available online at <<http://cnx.org/content/m10496/2.21/>>.

Using this and the fact that \mathcal{H} is linear, calculating $y(t)$ for combinations of complex exponentials is also straightforward. This linearity property is depicted in the two equations below - showing the input to the linear system H on the left side and the output, $y(t)$, on the right:

1.

$$c_1 e^{s_1 t} + c_2 e^{s_2 t} \rightarrow c_1 H(s_1) e^{s_1 t} + c_2 H(s_2) e^{s_2 t}$$

2.

$$\sum_n (c_n e^{s_n t}) \rightarrow \sum_n (c_n H(s_n) e^{s_n t})$$

The action of H on an input such as those in the two equations above is easy to explain: \mathcal{H} *independently* scales each exponential component $e^{s_n t}$ by a different complex number $H(s_n) \in \mathbb{C}$. As such, if we can write a function $f(t)$ as a combination of complex exponentials it allows us to:

- easily calculate the output of \mathcal{H} given $f(t)$ as an input (provided we know the eigenvalues $H(s)$)
- interpret how \mathcal{H} manipulates $f(t)$

6.2.2 Fourier Series

Joseph Fourier³ demonstrated that an arbitrary T-periodic function (Section 6.1) $f(t)$ can be written as a linear combination of harmonic complex sinusoids

$$f(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 n t}) \quad (6.3)$$

where $\omega_0 = \frac{2\pi}{T}$ is the fundamental frequency. For almost all $f(t)$ of practical interest, there exists c_n to make (6.3) true. If $f(t)$ is finite energy ($f(t) \in L^2[0, T]$), then the equality in (6.3) holds in the sense of energy convergence; if $f(t)$ is continuous, then (6.3) holds pointwise. Also, if $f(t)$ meets some mild conditions (the Dirichlet conditions), then (6.3) holds pointwise everywhere except at points of discontinuity.

The c_n - called the **Fourier coefficients** - tell us "how much" of the sinusoid $e^{j\omega_0 n t}$ is in $f(t)$. (6.3) essentially breaks down $f(t)$ into pieces, each of which is easily processed by an LTI system (since it is an eigenfunction of every LTI system). Mathematically, (6.3) tells us that the set of harmonic complex exponentials $\{e^{j\omega_0 n t}, n \in \mathbb{Z}\}$ form a basis for the space of T-periodic continuous time functions. Below are a few examples that are intended to help you think about a given signal or function, $f(t)$, in terms of its exponential basis functions.

6.2.2.1 Examples

For each of the given functions below, break it down into its "simpler" parts and find its fourier coefficients. Click to see the solution.

Exercise 6.1

(Solution on p. 139.)

$$f(t) = \cos(\omega_0 t)$$

Exercise 6.2

(Solution on p. 139.)

$$f(t) = \sin(2\omega_0 t)$$

Exercise 6.3

(Solution on p. 139.)

$$f(t) = 3 + 4\cos(\omega_0 t) + 2\cos(2\omega_0 t)$$

³<http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Fourier.html>

6.2.3 Fourier Coefficients

In general $f(t)$, the Fourier coefficients can be calculated from (6.3) by solving for c_n , which requires a little algebraic manipulation (for the complete derivation see the Fourier coefficients derivation (Section 6.3)). The end results will yield the following general equation for the fourier coefficients:

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-j\omega_0 nt} dt \quad (6.4)$$

The sequence of complex numbers $\{c_n, n \in \mathbb{Z}\}$ is just an alternate representation of the function $f(t)$. Knowing the Fourier coefficients c_n is the same as knowing $f(t)$ explicitly and vice versa. Given a periodic function, we can **transform** it into its Fourier series representation using (6.4). Likewise, we can **inverse transform** a given sequence of complex numbers, c_n , using (6.3) to reconstruct the function $f(t)$.

Along with being a natural representation for signals being manipulated by LTI systems, the Fourier series provides a description of periodic signals that is convenient in many ways. By looking at the Fourier series of a signal $f(t)$, we can infer mathematical properties of $f(t)$ such as smoothness, existence of certain symmetries, as well as the physically meaningful frequency content.

6.2.3.1 Example: Using Fourier Coefficient Equation

Here we will look at a rather simple example that almost requires the use of (6.4) to solve for the fourier coefficients. Once you understand the formula, the solution becomes a straightforward calculus problem. Find the fourier coefficients for the following equation:

Exercise 6.4

(Solution on p. 139.)

$$f(t) = \begin{cases} 1 & \text{if } |t| \leq T \\ 0 & \text{otherwise} \end{cases}$$

6.2.4 Summary: Fourier Series Equations

Our first equation (6.3) is the **synthesis** equation, which builds our function, $f(t)$, by combining sinusoids.

Synthesis

$$f(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 nt}) \quad (6.5)$$

And our second equation (6.4), termed the **analysis** equation, reveals how much of each sinusoid is in $f(t)$.

Analysis

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-j\omega_0 nt} dt \quad (6.6)$$

where we have stated that $\omega_0 = \frac{2\pi}{T}$.

NOTE: Understand that our interval of integration does not have to be $[0, T]$ in our Analysis Equation. We could use any interval $[a, a + T]$ of length T .

Example 6.1

This demonstration lets you synthesize a signal by combining sinusoids, similar to the synthesis equation for the Fourier series. See here⁴ for instructions on how to use the demo.

⁴"How to use the LabVIEW demos" <<http://cnx.org/content/m11550/latest/>>

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10496/latest/FourierCompManip.llb>

6.3 Derivation of Fourier Coefficients Equation⁵

6.3.1 Introduction

You should already be familiar with the existence of the general Fourier Series equation (Section 6.2.2: Fourier Series), which is written as:

$$f(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 n t}) \quad (6.7)$$

What we are interested in here is how to determine the Fourier coefficients, c_n , given a function $f(t)$. Below we will walk through the steps of deriving the general equation for the Fourier coefficients of a given function.

6.3.2 Derivation

To solve (6.7) for c_n , we have to do a little algebraic manipulation. First of all we will multiply both sides of (6.7) by $e^{-(j\omega_0 k t)}$, where $k \in \mathbb{Z}$.

$$f(t) e^{-(j\omega_0 k t)} = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 n t} e^{-(j\omega_0 k t)}) \quad (6.8)$$

Now integrate both sides over a given period, T :

$$\int_0^T f(t) e^{-(j\omega_0 k t)} dt = \int_0^T \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 n t} e^{-(j\omega_0 k t)}) dt \quad (6.9)$$

On the right-hand side we can switch the summation and integral along with pulling out the constant out of the integral.

$$\int_0^T f(t) e^{-(j\omega_0 k t)} dt = \sum_{n=-\infty}^{\infty} \left(c_n \int_0^T e^{j\omega_0 (n-k)t} dt \right) \quad (6.10)$$

Now that we have made this seemingly more complicated, let us focus on just the integral, $\int_0^T e^{j\omega_0 (n-k)t} dt$, on the right-hand side of the above equation. For this integral we will need to consider two cases: $n = k$ and $n \neq k$. For $n = k$ we will have:

$$\int_0^T e^{j\omega_0 (n-k)t} dt = T, \quad n = k \quad (6.11)$$

For $n \neq k$, we will have:

$$\int_0^T e^{j\omega_0 (n-k)t} dt = \int_0^T \cos(\omega_0 (n-k)t) dt + j \int_0^T \sin(\omega_0 (n-k)t) dt, \quad n \neq k \quad (6.12)$$

⁵This content is available online at <<http://cnx.org/content/m10733/2.6/>>.

But $\cos(\omega_0(n-k)t)$ has an integer number of periods, $n-k$, between 0 and T . Imagine a graph of the cosine; because it has an integer number of periods, there are equal areas above and below the x-axis of the graph. This statement holds true for $\sin(\omega_0(n-k)t)$ as well. What this means is

$$\int_0^T \cos(\omega_0(n-k)t) dt = 0 \quad (6.13)$$

as well as the integral involving the sine function. Therefore, we conclude the following about our integral of interest:

$$\int_0^T e^{j\omega_0(n-k)t} dt = \begin{cases} T & \text{if } n = k \\ 0 & \text{otherwise} \end{cases} \quad (6.14)$$

Now let us return our attention to our complicated equation, (6.10), to see if we can finish finding an equation for our Fourier coefficients. Using the facts that we have just proven above, we can see that the only time (6.10) will have a nonzero result is when k and n are equal:

$$\int_0^T f(t) e^{-j\omega_0 nt} dt = Tc_n, \quad n = k \quad (6.15)$$

Finally, we have our general equation for the Fourier coefficients:

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-j\omega_0 nt} dt \quad (6.16)$$

6.3.2.1 Finding Fourier Coefficients Steps

To find the Fourier coefficients of periodic $f(t)$:

1. For a given k , multiply $f(t)$ by $e^{-j\omega_0 kt}$, and take the area under the curve (dividing by T).
2. Repeat step (1) for all $k \in \mathbb{Z}$.

6.4 Fourier Series in a Nutshell⁶

6.4.1 Introduction

The convolution integral (Section 3.2) is the fundamental expression relating the input and output of an LTI system. However, it has three shortcomings:

1. It can be tedious to calculate.
2. It offers only limited physical interpretation of what the system is actually doing.
3. It gives little insight on how to design systems to accomplish certain tasks.

The Fourier Series (Section 6.2), along with the Fourier Transform and Laplace Transform, provides a way to address these three points. Central to all of these methods is the concept of an eigenfunction (Section 5.5) (or eigenvector (Section 5.3)). We will look at how we can rewrite any given signal, $f(t)$, in terms of complex exponentials (Section 1.6).

In fact, by making our notions of signals and linear systems more mathematically abstract, we will be able to draw enlightening parallels between signals and systems and linear algebra (Section 5.1).

⁶This content is available online at <http://cnx.org/content/m10751/2.3/>.

6.4.2 Eigenfunctions and LTI Systems

The action of a LTI system $\mathcal{H}[\dots]$ on one of its eigenfunctions e^{st} is

1. extremely easy (and fast) to calculate

$$\mathcal{H}[st] = H[s]e^{st} \tag{6.17}$$
2. easy to interpret: $\mathcal{H}[\dots]$ just *scales* e^{st} , keeping its frequency constant.

If only every function were an eigenfunction of $\mathcal{H}[\dots]$...

6.4.2.1 LTI System

... of course, not every function can be, but for LTI systems, their eigenfunctions span (Section 5.1.2: Span) the space of periodic functions (Section 6.1), meaning that for (almost) any periodic function $f(t)$ we can find $\{c_n\}$ where $n \in \mathbb{Z}$ and $c_i \in \mathbb{C}$ such that:

$$f(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 n t}) \tag{6.18}$$

Given (6.18), we can rewrite $\mathcal{H}[t] = y(t)$ as the following system

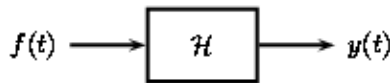


Figure 6.4: Transfer Functions modeled as LTI System.

where we have:

$$f(t) = \sum_n (c_n e^{j\omega_0 n t})$$

$$y(t) = \sum_n (c_n H(j\omega_0 n) e^{j\omega_0 n t})$$

This transformation from $f(t)$ to $y(t)$ can also be illustrated through the process below. Note that each arrow indicates an operation on our signal or coefficients.

$$f(t) \rightarrow \{c_n\} \rightarrow \{c_n H(j\omega_0 n)\} \rightarrow y(t) \tag{6.19}$$

where the three steps (arrows) in the above illustration represent the following three operations:

1. Transform with analysis (Fourier Coefficient (Section 6.3) equation):

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-j\omega_0 n t} dt$$

2. Action of \mathcal{H} on the Fourier series (Section 6.2) - equals a multiplication by $H(j\omega_0 n)$
3. Translate back to old basis - inverse transform using our synthesis equation from the Fourier series:

$$y(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 n t})$$

6.4.3 Physical Interpretation of Fourier Series

The Fourier series $\{c_n\}$ of a signal $f(t)$, defined in (6.18), also has a very important physical interpretation. Coefficient c_n tells us "how much" of frequency $\omega_0 n$ is in the signal.

Signals that vary slowly over time - **smooth signals** - have large c_n for small n .

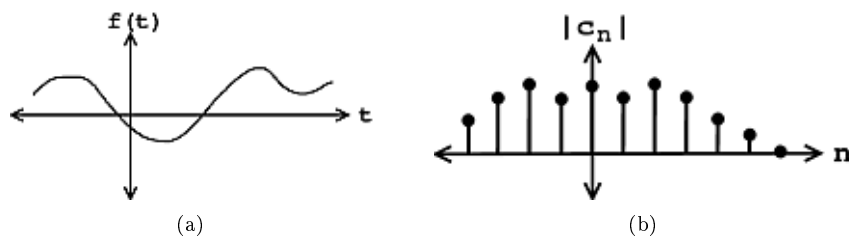


Figure 6.5: We begin with our smooth signal $f(t)$ on the left, and then use the Fourier series to find our Fourier coefficients - shown in the figure on the right.

Signals that vary quickly with time - **edgy** or **noisy signals** - will have large c_n for large n .

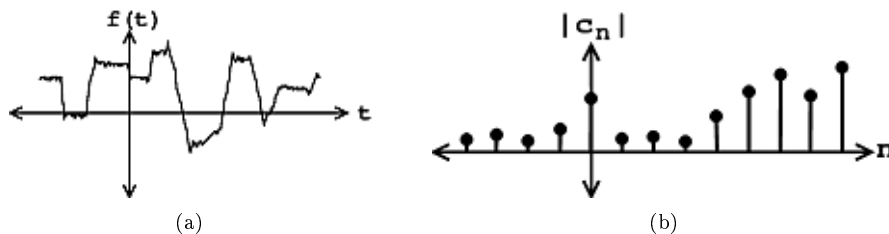


Figure 6.6: We begin with our noisy signal $f(t)$ on the left, and then use the Fourier series to find our Fourier coefficients - shown in the figure on the right.

Example 6.2: Periodic Pulse

We have the following pulse function, $f(t)$, over the interval $[-\frac{T}{2}, \frac{T}{2}]$:

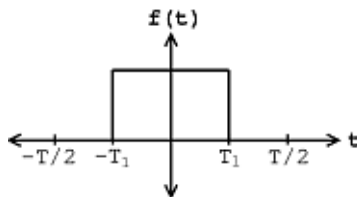


Figure 6.7: Periodic Signal $f(t)$

Using our formula for the Fourier coefficients,

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-j\omega_0 n t} dt \quad (6.20)$$

we can easily calculate our c_n . We will leave the calculation as an exercise for you! After solving the the equation for our $f(t)$, you will get the following results:

$$c_n = \begin{cases} \frac{2T_1}{T} & \text{if } n = 0 \\ \frac{2\sin(\omega_0 n T_1)}{n\pi} & \text{if } n \neq 0 \end{cases} \quad (6.21)$$

For $T_1 = \frac{T}{8}$, see the figure below for our results:

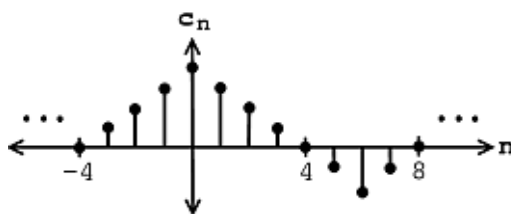


Figure 6.8: Our Fourier coefficients when $T_1 = \frac{T}{8}$

Our signal $f(t)$ is flat except for two edges (discontinuities). Because of this, c_n around $n = 0$ are large and c_n gets smaller as n approaches infinity.

QUESTION: Why does $c_n = 0$ for $n = \{\dots, -4, 4, 8, 16, \dots\}$? (What part of $e^{-(j\omega_0 n t)}$ lies over the pulse for these values of n ?)

6.5 Fourier Series Properties⁷

We will begin by refreshing your memory of our basic Fourier series (Section 6.2) equations:

$$f(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 n t}) \quad (6.22)$$

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-j\omega_0 n t} dt \quad (6.23)$$

Let $\mathcal{F}\{\cdot\}$ denote the transformation from $f(t)$ to the Fourier coefficients

$$\mathcal{F}\{f(t)\} = c_n \quad , \quad n \in \mathbb{Z}$$

$\mathcal{F}\{\cdot\}$ maps complex valued functions to sequences of complex numbers⁸.

⁷This content is available online at <<http://cnx.org/content/m10740/2.7/>>.

⁸"Complex Numbers" <<http://cnx.org/content/m0081/latest/>>

6.5.1 Linearity

$\mathcal{F}\{\cdot\}$ is a **linear transformation**.

Theorem 6.1:

If $\mathcal{F}\{f(t)\} = c_n$ and $\mathcal{F}\{g(t)\} = d_n$. Then

$$\mathcal{F}\{\alpha f(t)\} = \alpha c_n \quad , \quad \alpha \in \mathbb{C}$$

and

$$\mathcal{F}\{f(t) + g(t)\} = c_n + d_n$$

Proof: Easy. Just linearity of integral.

$$\begin{aligned} \mathcal{F}\{f(t) + g(t)\} &= \int_0^T (f(t) + g(t)) e^{-j\omega_0 n t} dt \quad , \quad n \in \mathbb{Z} \\ &= \frac{1}{T} \int_0^T f(t) e^{-j\omega_0 n t} dt + \frac{1}{T} \int_0^T g(t) e^{-j\omega_0 n t} dt \quad , \quad n \in \mathbb{Z} \\ &= c_n + d_n \quad , \quad n \in \mathbb{Z} \\ &= c_n + d_n \end{aligned} \tag{6.24}$$

6.5.2 Shifting

Shifting in time equals a phase shift of Fourier coefficients (Section 6.3)

Theorem 6.2:

$\mathcal{F}\{f(t - t_0)\} = e^{-j\omega_0 n t_0} c_n$ if $c_n = |c_n| e^{j\angle c_n}$, then

$$|e^{-j\omega_0 n t_0} c_n| = |e^{-j\omega_0 n t_0}| |c_n| = |c_n|$$

$$\angle e^{-j\omega_0 n t_0} c_n = \angle c_n - \omega_0 t_0 n$$

Proof:

$$\begin{aligned} \mathcal{F}\{f(t - t_0)\} &= \frac{1}{T} \int_0^T f(t - t_0) e^{-j\omega_0 n t} dt \quad , \quad n \in \mathbb{Z} \\ &= \frac{1}{T} \int_{-t_0}^{T-t_0} f(t - t_0) e^{-j\omega_0 n (t - t_0)} e^{-j\omega_0 n t_0} dt \quad , \quad n \in \mathbb{Z} \\ &= \frac{1}{T} \int_{-t_0}^{T-t_0} f(\tilde{t}) e^{-j\omega_0 n \tilde{t}} e^{-j\omega_0 n t_0} dt \quad , \quad n \in \mathbb{Z} \\ &= e^{-j\omega_0 n t_0} c_n \quad , \quad n \in \mathbb{Z} \end{aligned} \tag{6.25}$$

6.5.3 Parseval's Relation

$$\int_0^T (|f(t)|)^2 dt = T \sum_{n=-\infty}^{\infty} (|c_n|^2) \tag{6.26}$$

Parseval's relation allows us to calculate the energy of a signal from its Fourier series.

NOTE: Parseval tells us that the Fourier series maps $L^2([0, T])$ to $l^2(\mathbb{Z})$.

$$f(t) \xleftrightarrow{F.S.} \{c_n\}$$

$$f(t) \in L^2[0, T] \iff \{c_n\} \in l^2\{\mathbb{Z}\}$$

Figure 6.9

Exercise 6.5

For $f(t)$ to have "finite energy," what do the c_n do as $n \rightarrow \infty$?

(Solution on p. 140.)

Exercise 6.6

If $c_n = \frac{1}{n}$, $|n| > 0$, is $f \in L^2([0, T])$?

(Solution on p. 140.)

Exercise 6.7

Now, if $c_n = \frac{1}{\sqrt{n}}$, $|n| > 0$, is $f \in L^2([0, T])$?

(Solution on p. 140.)

The rate of decay of the Fourier series determines if $f(t)$ has *finite energy*.

6.5.4 Differentiation in Fourier Domain

$$\mathcal{F}\{f(t)\} = c_n \Rightarrow \mathcal{F}\left\{\frac{d}{dt}f(t)\right\} = jn\omega_0 c_n \quad (6.27)$$

Since

$$f(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 n t}) \quad (6.28)$$

then

$$\begin{aligned} \frac{d}{dt}f(t) &= \sum_{n=-\infty}^{\infty} (c_n \frac{d}{dt}(e^{j\omega_0 n t})) \\ &= \sum_{n=-\infty}^{\infty} (c_n j\omega_0 n e^{j\omega_0 n t}) \end{aligned} \quad (6.29)$$

A differentiator *attenuates* the low frequencies in $f(t)$ and *accentuates* the high frequencies. It removes general trends and accentuates areas of sharp variation.

NOTE: A common way to mathematically measure the smoothness of a function $f(t)$ is to see how many derivatives are finite energy.

This is done by looking at the Fourier coefficients of the signal, specifically how fast they *decay* as $n \rightarrow \infty$. If $\mathcal{F}\{f(t)\} = c_n$ and $|c_n|$ has the form $\frac{1}{n^k}$, then $\mathcal{F}\left\{\frac{d^m}{dt^m}f(t)\right\} = (jn\omega_0)^m c_n$ and has the form $\frac{n^m}{n^k}$. So for the m^{th} derivative to have finite energy, we need

$$\sum \left(\left(\left| \frac{n^m}{n^k} \right| \right)^2 \right) < \infty$$

thus $\frac{n^m}{n^k}$ decays *faster* than $\frac{1}{n}$ which implies that

$$2k - 2m > 1$$

or

$$k > \frac{2m + 1}{2}$$

Thus the decay rate of the Fourier series dictates smoothness.

6.5.5 Integration in the Fourier Domain

If

$$\mathcal{F}\{f(t)\} = c_n \tag{6.30}$$

then

$$\mathcal{F}\left\{\int_{-\infty}^t f(\tau) d\tau\right\} = \frac{1}{j\omega_0 n} c_n \tag{6.31}$$

NOTE: If $c_0 \neq 0$, this expression doesn't make sense.

Integration accentuates low frequencies and attenuates high frequencies. Integrators bring out the *general trends* in signals and suppress short term variation (which is noise in many cases). Integrators are *much* nicer than differentiators.

6.5.6 Signal Multiplication

Given a signal $f(t)$ with Fourier coefficients c_n and a signal $g(t)$ with Fourier coefficients d_n , we can define a new signal, $y(t)$, where $y(t) = f(t)g(t)$. We find that the Fourier Series representation of $y(t)$, e_n , is such that $e_n = \sum_{k=-\infty}^{\infty} (c_k d_{n-k})$. This is to say that signal multiplication in the time domain is equivalent to discrete-time convolution (Section 4.2) in the frequency domain. The proof of this is as follows

$$\begin{aligned} e_n &= \frac{1}{T} \int_0^T f(t) g(t) e^{-j\omega_0 n t} dt \\ &= \frac{1}{T} \int_0^T \left(\sum_{k=-\infty}^{\infty} (c_k e^{j\omega_0 k t}) \right) g(t) e^{-j\omega_0 n t} dt \\ &= \sum_{k=-\infty}^{\infty} \left(c_k \left(\frac{1}{T} \int_0^T g(t) e^{-j\omega_0 (n-k)t} dt \right) \right) \\ &= \sum_{k=-\infty}^{\infty} (c_k d_{n-k}) \end{aligned} \tag{6.32}$$

6.6 Symmetry Properties of the Fourier Series⁹

6.6.1 Symmetry Properties

6.6.1.1 Real Signals

Real signals have a conjugate symmetric Fourier series.

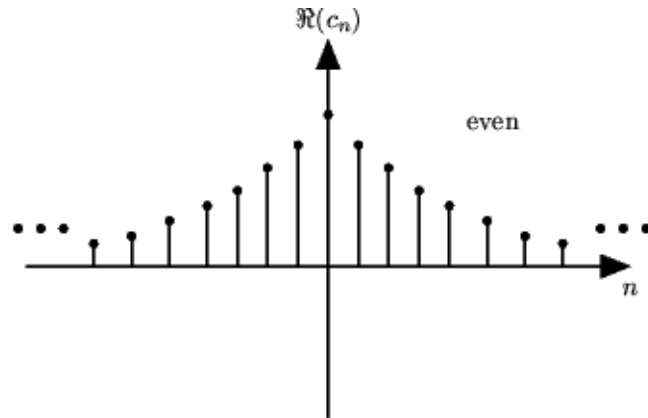
Theorem 6.3:

If $f(t)$ is real it implies that $f(t) = f(t)^*$ ($f(t)^*$ is the complex conjugate of $f(t)$), then $c_n = c_{-n}^*$ which implies that $Re(c_n) = Re(c_{-n})$, i.e. the real part of c_n is even, and $Im(c_n) = -(Im(c_{-n}))$, i.e. the imaginary part of c_n is odd. See Figure 6.10. It also implies that $|c_n| = |c_{-n}|$, i.e. that magnitude is even, and that $\angle c_n = (\angle -c_{-n})$, i.e. the phase is odd.

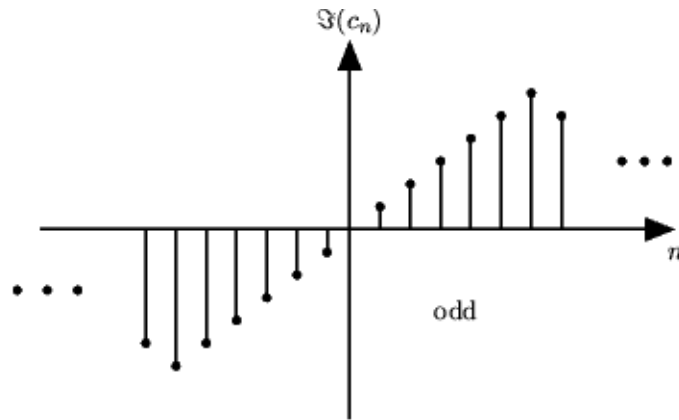
⁹This content is available online at <<http://cnx.org/content/m10838/2.4/>>.

Proof:

$$\begin{aligned}
 c_{-n} &= \frac{1}{T} \int_0^T f(t) e^{j\omega_0 n t} dt \\
 &= \frac{1}{T} \int_0^T f(t)^* e^{-(j\omega_0 n t)} dt^* \quad , \quad f(t) = f(t)^* \\
 &= \frac{1}{T} \int_0^T f(t) e^{-(j\omega_0 n t)} dt^* \\
 &= c_n^*
 \end{aligned}
 \tag{6.33}$$



(a)



(b)

Figure 6.10: $Re(c_n) = Re(c_{-n})$, and $Im(c_n) = -(Im(c_{-n}))$.

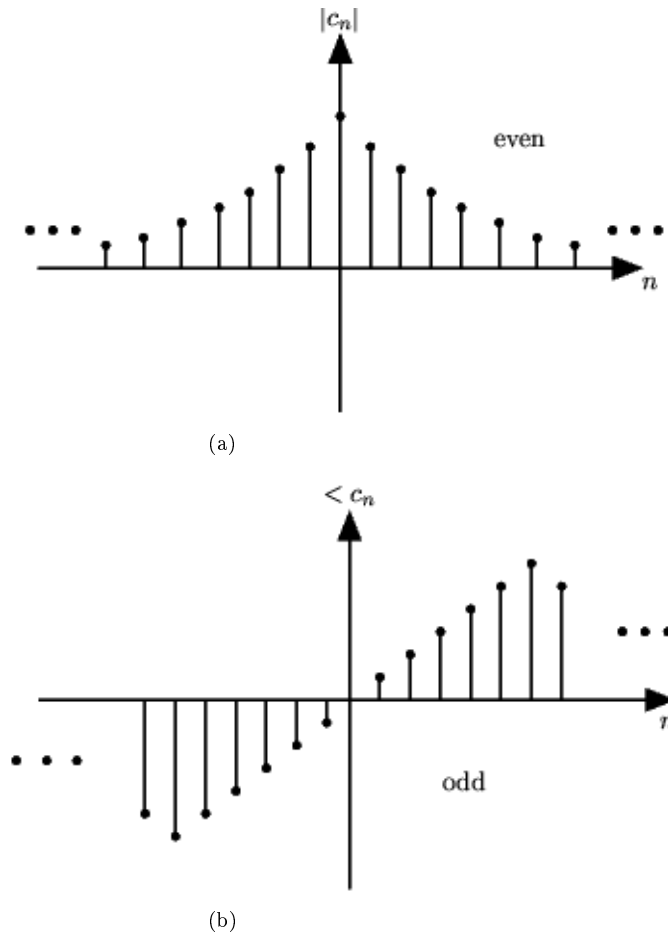


Figure 6.11: $|c_n| = |c_{-n}|$, and $\angle c_n = (\angle -c_{-n})$.

6.6.1.2 Real and Even Signals

Real and even signals have real and even Fourier series.

Theorem 6.4:

If $f(t) = f(t)^*$ and $f(t) = f(-t)$, i.e. the signal is real and even, then $c_n = c_{-n}$ and $c_n = c_n^*$.

Proof:

$$\begin{aligned}
 c_n &= \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-j\omega_0 n t} dt \\
 &= \frac{1}{T} \int_{-\frac{T}{2}}^0 f(t) e^{-j\omega_0 n t} dt + \frac{1}{T} \int_0^{\frac{T}{2}} f(t) e^{-j\omega_0 n t} dt \\
 &= \frac{1}{T} \int_0^{\frac{T}{2}} f(-t) e^{j\omega_0 n t} dt + \frac{1}{T} \int_0^{\frac{T}{2}} f(t) e^{-j\omega_0 n t} dt \\
 &= \frac{2}{T} \int_0^{\frac{T}{2}} f(t) \cos(\omega_0 n t) dt
 \end{aligned} \tag{6.34}$$

$f(t)$ and $\cos(\omega_0 n t)$ are both real which implies that c_n is real. Also $\cos(\omega_0 n t) = \cos(-(\omega_0 n t))$ so $c_n = c_{-n}$. It is also easy to show that $f(t) = 2 \sum_{n=0}^{\infty} (c_n \cos(\omega_0 n t))$ since $f(t)$, c_n , and $\cos(\omega_0 n t)$

are all real and even.

6.6.1.3 Real and Odd Signals

Real and odd signals have Fourier Series that are odd and purely imaginary.

Theorem 6.5:

If $f(t) = -(f(-t))$ and $f(t) = f(t)^*$, i.e. the signal is real and odd, then $c_n = -c_{-n}$ and $c_n = -(c_n^*)$, i.e. c_n is odd and purely imaginary.

Proof: Do it at home.

If $f(t)$ is odd, then we can expand it in terms of $\sin(\omega_0 nt)$:

$$f(t) = \sum_{n=1}^{\infty} (2c_n \sin(\omega_0 nt))$$

6.6.2 Summary

In summary, we can find $f_e(t)$, an even function, and $f_o(t)$, an odd function, such that

$$f(t) = f_e(t) + f_o(t) \tag{6.35}$$

which implies that, for any $f(t)$, we can find $\{a_n\}$ and $\{b_n\}$ such that

$$f(t) = \sum_{n=0}^{\infty} (a_n \cos(\omega_0 nt)) + \sum_{n=1}^{\infty} (b_n \sin(\omega_0 nt)) \tag{6.36}$$

Example 6.3: Triangle Wave

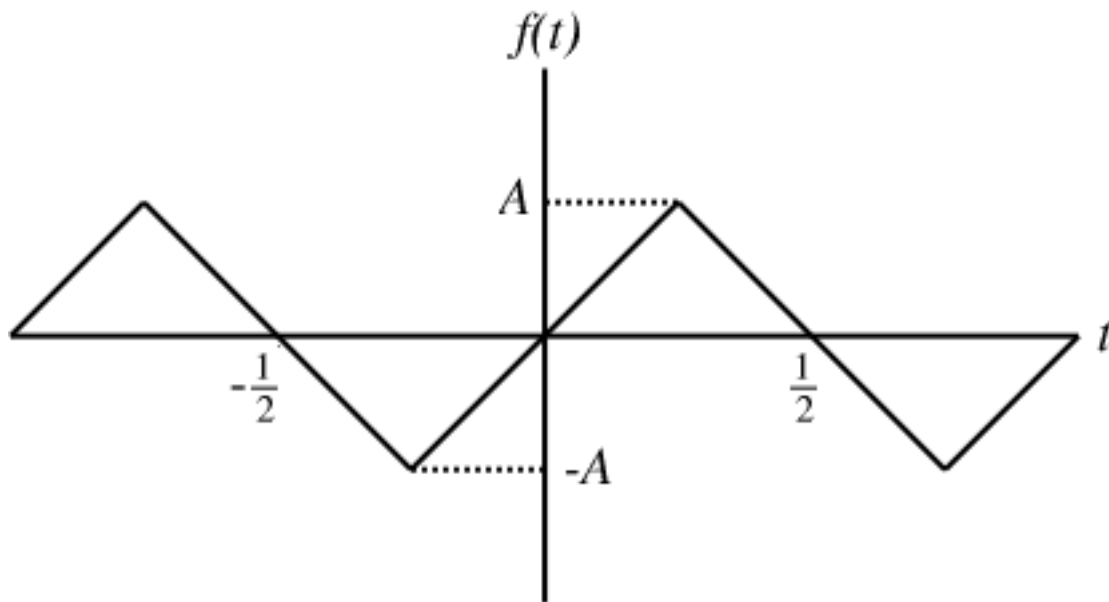


Figure 6.12: $T = 1$ and $\omega_0 = 2\pi$.

$f(t)$ is real and odd.

$$c_n = \begin{cases} \frac{4A}{j\pi^2 n^2} & \text{if } n = \{\dots, -11, -7, -3, 1, 5, 9, \dots\} \\ -\left(\frac{4A}{j\pi^2 n^2}\right) & \text{if } n = \{\dots, -9, -5, -1, 3, 7, 11, \dots\} \\ 0 & \text{if } n = \{\dots, -4, -2, 0, 2, 4, \dots\} \end{cases}$$

Does $c_n = -c_{-n}$?

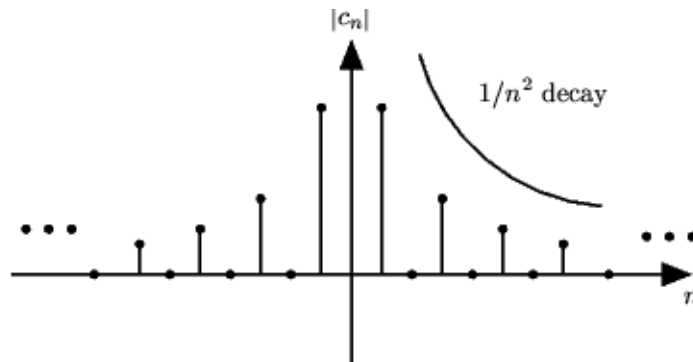


Figure 6.13: The Fourier series of a triangle wave.

NOTE: We can often gather information about the *smoothness* of a signal by examining its Fourier coefficients.

Take a look at the above examples. The pulse and sawtooth waves are not continuous and their Fourier series' fall off like $\frac{1}{n}$. The triangle wave is continuous, but not differentiable and its Fourier series falls off like $\frac{1}{n^2}$.

The next 3 properties will give a better feel for this.

6.7 Circular Convolution Property of Fourier Series¹⁰

6.7.1 Signal Circular Convolution

Given a signal $f(t)$ with Fourier coefficients c_n and a signal $g(t)$ with Fourier coefficients d_n , we can define a new signal, $v(t)$, where $v(t) = (f(t) \otimes g(t))$. We find that the Fourier Series (Section 6.2) representation of $y(t)$, a_n , is such that $a_n = c_n d_n$. $(f(t) \otimes g(t))$ is the circular convolution (Section 4.3) of two periodic signals and is equivalent to the convolution over one interval, i.e. $(f(t) \otimes g(t)) = \int_0^T \int_0^T f(\tau) g(t - \tau) d\tau dt$.

NOTE: Circular convolution in the time domain is equivalent to multiplication of the Fourier coefficients.

¹⁰This content is available online at <http://cnx.org/content/m10839/2.4/>.

This is proved as follows

$$\begin{aligned}
 a_n &= \frac{1}{T} \int_0^T v(t) e^{-j\omega_0 n t} dt \\
 &= \frac{1}{T^2} \int_0^T \int_0^T f(\tau) g(t-\tau) d\tau e^{-j\omega_0 n t} dt \\
 &= \frac{1}{T} \int_0^T f(\tau) \left(\frac{1}{T} \int_0^T g(t-\tau) e^{-j\omega_0 n t} dt \right) d\tau \\
 &= \frac{1}{T} \int_0^T f(\tau) \left(\frac{1}{T} \int_{-\tau}^{T-\tau} g(\nu) e^{-j\omega_0(\nu+\tau)} d\nu \right) d\tau, \quad \nu = t - \tau \\
 &= \frac{1}{T} \int_0^T f(\tau) \left(\frac{1}{T} \int_{-\tau}^{T-\tau} g(\nu) e^{-j\omega_0 n \nu} d\nu \right) e^{-j\omega_0 n \tau} d\tau \\
 &= \frac{1}{T} \int_0^T f(\tau) d_n e^{-j\omega_0 n \tau} d\tau \\
 &= d_n \left(\frac{1}{T} \int_0^T f(\tau) e^{-j\omega_0 n \tau} d\tau \right) \\
 &= c_n d_n
 \end{aligned} \tag{6.37}$$

Example 6.4

Take a look at a square pulse with a period, $T_1 = \frac{T}{4}$:

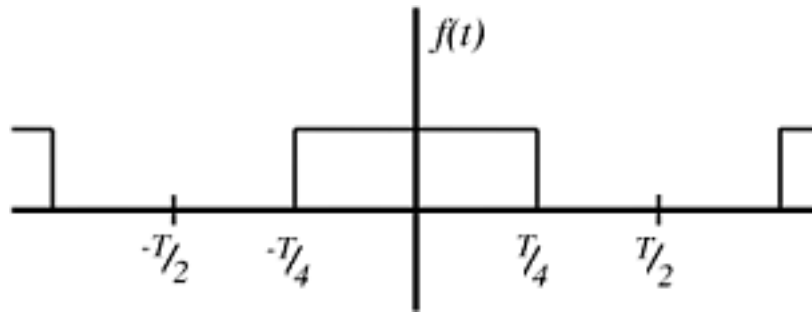


Figure 6.14

For this signal

$$c_n = \begin{cases} \frac{1}{T} & \text{if } n = 0 \\ \frac{1}{2} \frac{\sin(\frac{\pi}{2}n)}{\frac{\pi}{2}n} & \text{otherwise} \end{cases}$$

Exercise 6.8

(Solution on p. 140.)

What signal has Fourier coefficients $a_n = c_n^2 = \frac{1}{4} \frac{\sin^2(\frac{\pi}{2}n)}{(\frac{\pi}{2}n)^2}$?

6.8 Fourier Series and LTI Systems¹¹

6.8.1 Introducing the Fourier Series to LTI Systems

Before looking at this module, one should be familiar with the concepts of eigenfunction and LTI systems (Section 5.5). Recall, for \mathcal{H} LTI system we get the following relationship

¹¹This content is available online at <<http://cnx.org/content/m10752/2.7/>>.

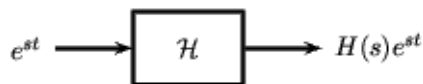


Figure 6.15: Input and output signals to our LTI system.

where e^{st} is an eigenfunction of \mathcal{H} . Its corresponding eigenvalue (Section 5.2) $H(s)$ can be calculated using the impulse response (Section 1.5) $h(t)$

$$H(s) = \int_{-\infty}^{\infty} h(\tau) e^{-(s\tau)} d\tau$$

So, using the Fourier Series (Section 6.2) expansion for periodic (Section 6.1) $f(t)$ where we input

$$f(t) = \sum_n (c_n e^{j\omega_0 n t})$$

into the system,

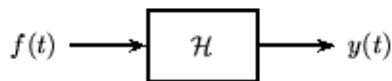


Figure 6.16: LTI system

our output $y(t)$ will be

$$y(t) = \sum_n (H(j\omega_0 n) c_n e^{j\omega_0 n t})$$

So we can see that by applying the Fourier series expansion equations, we can go from $f(t)$ to c_n and vice versa, and we do the same for our output, $y(t)$

6.8.2 Effects of Fourier Series

We can think of an LTI system as *shaping* the frequency content of the input. Keep in mind the basic LTI system we presented above in Figure 6.16. The LTI system, \mathcal{H} , simply multiplies all of our Fourier coefficients and scales them.

Given the Fourier coefficients $\{c_n\}$ of the input and the eigenvalues of the system $\{H(j\omega_0 n)\}$, the Fourier series of the output is $\{H(j\omega_0 n) c_n\}$ (simple term-by-term multiplication).

NOTE: The eigenvalues $H(j\omega_0 n)$ completely describe what a LTI system does to periodic signals with period $T = 2\pi\omega_0$

Example 6.5

What does this system do?

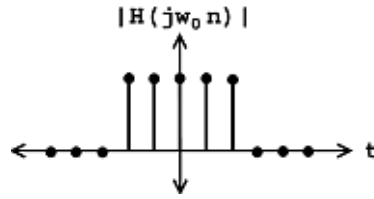


Figure 6.17

Example 6.6

What about this system?

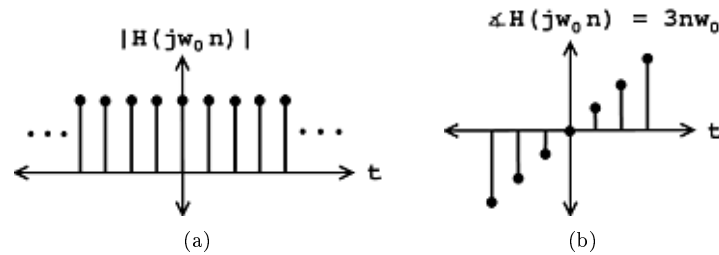


Figure 6.18

6.8.3 Examples**Example 6.7: RC Circuit**

$$h(t) = \frac{1}{RC} e^{-\frac{t}{RC}} u(t)$$

What does this system do to the Fourier Series of an input $f(t)$?
Calculate the eigenvalues of this system

$$\begin{aligned}
 H(s) &= \int_{-\infty}^{\infty} h(\tau) e^{-s\tau} d\tau \\
 &= \int_0^{\infty} \frac{1}{RC} e^{-\frac{\tau}{RC}} e^{-s\tau} d\tau \\
 &= \frac{1}{RC} \int_0^{\infty} e^{(-\tau)(\frac{1}{RC}+s)} d\tau \\
 &= \frac{1}{RC} \frac{1}{\frac{1}{RC}+s} e^{(-\tau)(\frac{1}{RC}+s)} \Big|_{\tau=0}^{\infty} \\
 &= \frac{1}{1+RCs}
 \end{aligned} \tag{6.38}$$

Now, say we feed the RC circuit a periodic (period $T = 2\pi w_0$) input $f(t)$.

Look at the eigenvalues for $s = jw_0n$

$$|H(jw_0n)| = \frac{1}{|1 + RCjw_0n|} = \frac{1}{\sqrt{1 + R^2C^2w_0^2n^2}}$$

The RC circuit is a *lowpass* system: it passes low frequencies (n around 0) and attenuates high frequencies (large n).

Example 6.8: Square pulse wave through RC circuit

- Input Signal: Taking the fourier series of $f(t)$

$$c_n = \frac{1}{2} \frac{\sin\left(\frac{\pi}{2}n\right)}{\frac{\pi}{2}n}$$

- $\frac{1}{t}$ at $n = 0$
- System: eigenvalues

$$H(jw_0n) = \frac{1}{1 + jRCw_0n}$$

- Output Signal: Taking the fourier series of $y(t)$

$$d_n = H(jw_0n) c_n = \frac{1}{1 + jRCw_0n} \frac{1}{2} \frac{\sin\left(\frac{\pi}{2}n\right)}{\frac{\pi}{2}n}$$

$$d_n = \frac{1}{1 + jRCw_0n} \frac{1}{2} \frac{\sin\left(\frac{\pi}{2}n\right)}{\frac{\pi}{2}n}$$

$$y(t) = \sum (d_n e^{jw_0nt})$$

What can we infer about $y(t)$ from $\{d_n\}$?

1. Is $y(t)$ real?
2. Is $y(t)$ even symmetric? odd symmetric?
3. Qualitatively, what does $y(t)$ look like? Is it "smoother" than $f(t)$? (decay rate of d_n vs. c_n)

$$d_n = \frac{1}{1 + jRCw_0n} \frac{1}{2} \frac{\sin\left(\frac{\pi}{2}n\right)}{\frac{\pi}{2}n}$$

$$|d_n| = \frac{1}{\sqrt{1 + (RCw_0)^2n^2}} \frac{1}{2} \frac{\sin\left(\frac{\pi}{2}n\right)}{\frac{\pi}{2}n}$$

6.9 Convergence of Fourier Series¹²

6.9.1 Introduction

Before looking at this module, hopefully you have become fully convinced of the fact that any periodic function, $f(t)$, can be represented as a sum of complex sinusoids (Section 1.4). If you are not, then try looking back at eigen-stuff in a nutshell (Section 5.4) or eigenfunctions of LTI systems (Section 5.5). We

¹²This content is available online at <<http://cnx.org/content/m10745/2.3/>>.

have shown that we can represent a signal as the sum of exponentials through the Fourier Series (Section 6.2) equations below:

$$f(t) = \sum_n (c_n e^{j\omega_0 n t}) \quad (6.39)$$

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-(j\omega_0 n t)} dt \quad (6.40)$$

Joseph Fourier¹³ insisted that these equations were true, but could not prove it. Lagrange publicly ridiculed Fourier, and said that only continuous functions can be represented by (6.39) (indeed he proved that (6.39) holds for continuous-time functions). However, we know now that the real truth lies in between Fourier and Lagrange's positions.

6.9.2 Understanding the Truth

Formulating our question mathematically, let

$$f_{N'}(t) = \sum_{n=-N}^N (c_n e^{j\omega_0 n t})$$

where c_n equals the Fourier coefficients of $f(t)$ (see (6.40)).

$f_{N'}(t)$ is a "partial reconstruction" of $f(t)$ using the first $2N+1$ Fourier coefficients. $f_{N'}(t)$ approximates $f(t)$, with the approximation getting better and better as N gets large. Therefore, we can think of the set $\{f_{N'}(t), N = \{0, 1, \dots\}\}$ as a **sequence of functions**, each one approximating $f(t)$ better than the one before.

The question is, does this sequence converge to $f(t)$? Does $f_{N'}(t) \rightarrow f(t)$ as $N \rightarrow \infty$? We will try to answer this question by thinking about convergence in two different ways:

1. Looking at the **energy** of the error signal:

$$e_N(t) = f(t) - f_{N'}(t)$$

2. Looking at $\lim_{N \rightarrow \infty} f_{N'}(t)$ at each point and comparing to $f(t)$.

6.9.2.1 Approach #1

Let $e_N(t)$ be the difference (i.e. error) between the signal $f(t)$ and its partial reconstruction $f_{N'}(t)$

$$e_N(t) = f(t) - f_{N'}(t) \quad (6.41)$$

If $f(t) \in L^2([0, T])$ (finite energy), then the energy of $e_N(t) \rightarrow 0$ as $N \rightarrow \infty$ is

$$\int_0^T (|e_N(t)|)^2 dt = \int_0^T (f(t) - f_{N'}(t))^2 dt \rightarrow 0 \quad (6.42)$$

We can prove this equation using Parseval's relation:

$$\lim_{N \rightarrow \infty} \int_0^T (|f(t) - f_{N'}(t)|)^2 dt = \lim_{N \rightarrow \infty} \sum_{n=-\infty}^{\infty} \left((|\mathcal{F}_n f(t) - \mathcal{F}_n f_{N'}(t)|)^2 \right) = \lim_{N \rightarrow \infty} \sum_{|n| > N} (|c_n|^2) = 0$$

where the last equation before zero is the tail sum of the Fourier Series, which approaches zero because $f(t) \in L^2([0, T])$. Since physical systems respond to energy, the Fourier Series provides an adequate representation for all $f(t) \in L^2([0, T])$ equaling finite energy over one period.

¹³<http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Fourier.html>

6.9.2.2 Approach #2

The fact that $e_N \rightarrow 0$ says nothing about $f(t)$ and $\lim_{N \rightarrow \infty} f_N'(t)$ being *equal* at a given point. Take the two functions graphed below for example:

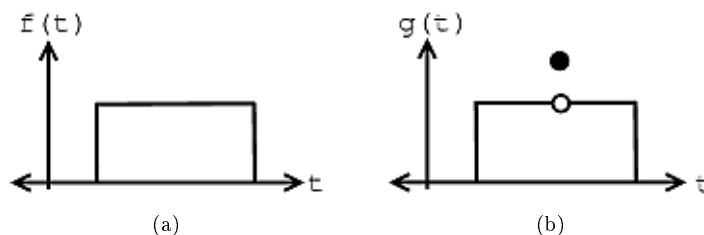


Figure 6.19

Given these two functions, $f(t)$ and $g(t)$, then we can see that for all t , $f(t) \neq g(t)$, but

$$\int_0^T (|f(t) - g(t)|)^2 dt = 0$$

From this we can see the following relationships:

$$\text{energyconvergence} \neq \text{pointwiseconvergence}$$

$$\text{pointwiseconvergence} \Rightarrow \text{convergencein}L^2([0, T])$$

However, the reverse of the above statement does not hold true.

It turns out that if $f(t)$ has a **discontinuity** (as can be seen in figure of $g(t)$ above) at t_0 , then

$$f(t_0) \neq \lim_{N \rightarrow \infty} f_N'(t_0)$$

But as long as $f(t)$ meets some other fairly mild conditions, then

$$f(t') = \lim_{N \rightarrow \infty} f_N'(t')$$

if $f(t)$ is **continuous** at $t = t'$.

6.10 Dirichlet Conditions¹⁴

Named after the German mathematician, Peter Dirichlet, the **Dirichlet conditions** are the sufficient conditions to guarantee **existence** and **convergence** of the Fourier series (Section 6.2) or the Fourier transform¹⁵.

¹⁴This content is available online at <<http://cnx.org/content/m10089/2.9/>>.

¹⁵"Derivation of the Fourier Transform" <<http://cnx.org/content/m0046/latest/>>

6.10.1 The Weak Dirichlet Condition for the Fourier Series

Condition 6.1: The Weak Dirichlet Condition

For the Fourier Series to exist, the Fourier coefficients must be finite. The **Weak Dirichlet Condition** guarantees this existence. It essentially says that the integral of the absolute value of the signal must be finite. The limits of integration are different for the Fourier Series case than for the Fourier Transform case. This is a direct result of the differing definitions of the two.

Proof: The Fourier Series exists (the coefficients are finite) if

Weak Dirichlet Condition for the Fourier Series

$$\int_0^T |f(t)| dt < \infty \quad (6.43)$$

This can be shown from the initial condition that the Fourier Series coefficients be finite.

$$|c_n| = \left| \frac{1}{T} \int_0^T f(t) e^{-j\omega_0 n t} dt \right| \leq \frac{1}{T} \int_0^T |f(t)| |e^{-j\omega_0 n t}| dt \quad (6.44)$$

Remembering our complex exponentials (Section 1.6), we know that in the above equation $|e^{-j\omega_0 n t}| = 1$, which gives us

$$\frac{1}{T} \int_0^T |f(t)| dt = \frac{1}{T} \int_0^T |f(t)| dt \quad (6.45)$$

$$< \infty \quad (6.46)$$

NOTE: If we have the function:

$$f(t) = \frac{1}{t}, \quad 0 < t \leq T$$

then you should note that this functions *fails* the above condition.

6.10.1.1 The Weak Dirichlet Condition for the Fourier Transform

Condition 6.2:

The Fourier Transform exists if

Weak Dirichlet Condition for the Fourier Transform

$$\int_{-\infty}^{\infty} |f(t)| dt < \infty \quad (6.47)$$

This can be derived the same way the weak Dirichlet for the Fourier Series was derived, by beginning with the definition and showing that the Fourier Transform must be less than infinity everywhere.

6.10.2 The Strong Dirichlet Conditions

The Fourier Transform exists if the signal has a finite number of discontinuities and a finite number of **maxima** and **minima**. For the Fourier Series to exist, the following two conditions must be satisfied (along with the Weak Dirichlet Condition):

1. In one period, $f(t)$ has only a finite number of minima and maxima.
2. In one period, $f(t)$ has only a finite number of discontinuities and each one is finite.

These are what we refer to as the **Strong Dirichlet Conditions**. In theory we can think of signals that violate these conditions, $\sin(\log t)$ for instance. However, it is not possible to create a signal that violates these conditions in a lab. Therefore, any real-world signal will have a Fourier representation.

6.10.2.1 Example

Let us assume we have the following function and equality:

$$f'(t) = \lim_{N \rightarrow \infty} f_N'(t) \quad (6.48)$$

If $f(t)$ meets all three conditions of the Strong Dirichlet Conditions, then

$$f(\tau) = f'(\tau)$$

at every τ at which $f(t)$ is continuous. And where $f(t)$ is discontinuous, $f'(t)$ is the average of the values on the right and left. See Figure 6.20 as an example:

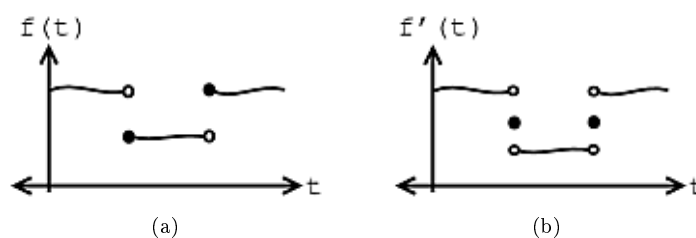


Figure 6.20: Discontinuous functions, $f(t)$.

NOTE: The functions that fail the Dirichlet conditions are pretty pathological - as engineers, we are not too interested in them.

6.11 Gibbs's Phenomena¹⁶

6.11.1 Introduction

The Fourier Series (Section 6.2) is the representation of continuous-time, periodic signals in terms of complex exponentials. The Dirichlet conditions (Section 6.10) suggest that discontinuous signals may have a Fourier Series representation so long as there are a finite number of discontinuities. This seems counter-intuitive, however, as complex exponentials (Section 1.6) are continuous functions. It does not seem possible to exactly reconstruct a discontinuous function from a set of continuous ones. In fact, it is not. However, it can be if we relax the condition of 'exactly' and replace it with the idea of 'almost everywhere'. This is to say that the reconstruction is exactly the same as the original signal except at a finite number of points. These points, not necessarily surprisingly, occur at the points of discontinuities.

6.11.1.1 History

In the late 1800s, many machines were built to calculate Fourier coefficients and re-synthesize:

$$f_N'(t) = \sum_{n=-N}^N (c_n e^{j\omega_0 n t}) \quad (6.49)$$

¹⁶This content is available online at <<http://cnx.org/content/m10092/2.9/>>.

Albert Michelson (an extraordinary experimental physicist) built a machine in 1898 that could compute c_n up to $n = \pm 79$, and he re-synthesized

$$f_{79}'(t) = \sum_{n=-79}^{79} (c_n e^{j\omega_0 n t}) \quad (6.50)$$

The machine performed very well on all tests except those involving **discontinuous functions**. When a square wave, like that shown in Figure 6.21 (Fourier series approximation of a square wave), was inputted into the machine, "wiggles" around the discontinuities appeared, and even as the number of Fourier coefficients approached infinity, the wiggles never disappeared - these can be seen in the last plot in Figure 6.21 (Fourier series approximation of a square wave). J. Willard Gibbs first explained this phenomenon in 1899, and therefore these discontinuous points are referred to as **Gibbs Phenomenon**.

6.11.2 Explanation

We begin this discussion by taking a signal with a finite number of discontinuities (like a **square pulse**) and finding its Fourier Series representation. We then attempt to reconstruct it from these Fourier coefficients. What we find is that the more coefficients we use, the more the signal begins to resemble the original. However, around the discontinuities, we observe rippling that does not seem to subside. As we consider even more coefficients, we notice that the ripples narrow, but do not shorten. As we approach an infinite number of coefficients, this rippling still does not go away. This is when we apply the idea of almost everywhere. While these ripples remain (never dropping below 9% of the pulse height), the area inside them tends to zero, meaning that the energy of this ripple goes to zero. This means that their width is approaching zero and we can assert that the reconstruction is exactly the original except at the points of discontinuity. Since the Dirichlet conditions assert that there may only be a finite number of discontinuities, we can conclude that the principle of almost everywhere is met. This phenomenon is a specific case of **nonuniform convergence**.

Below we will use the square wave, along with its Fourier Series representation, and show several figures that reveal this phenomenon more mathematically.

6.11.2.1 Square Wave

The Fourier series representation of a square signal below says that the left and right sides are "equal." In order to understand Gibbs Phenomenon we will need to redefine the way we look at equality.

$$s(t) = a_0 + \sum_{k=1}^{\infty} \left(a_k \cos \left(\frac{2\pi kt}{T} \right) \right) + \sum_{k=1}^{\infty} \left(b_k \sin \left(\frac{2\pi kt}{T} \right) \right) \quad (6.51)$$

6.11.2.2 Example

Figure 6.21 (Fourier series approximation of a square wave) shows several Fourier series approximation of the square wave¹⁷ using a varied number of terms, denoted by K :

¹⁷"Fourier Series Approximation of a Square Wave" <<http://cnx.org/content/m0041/latest/>>

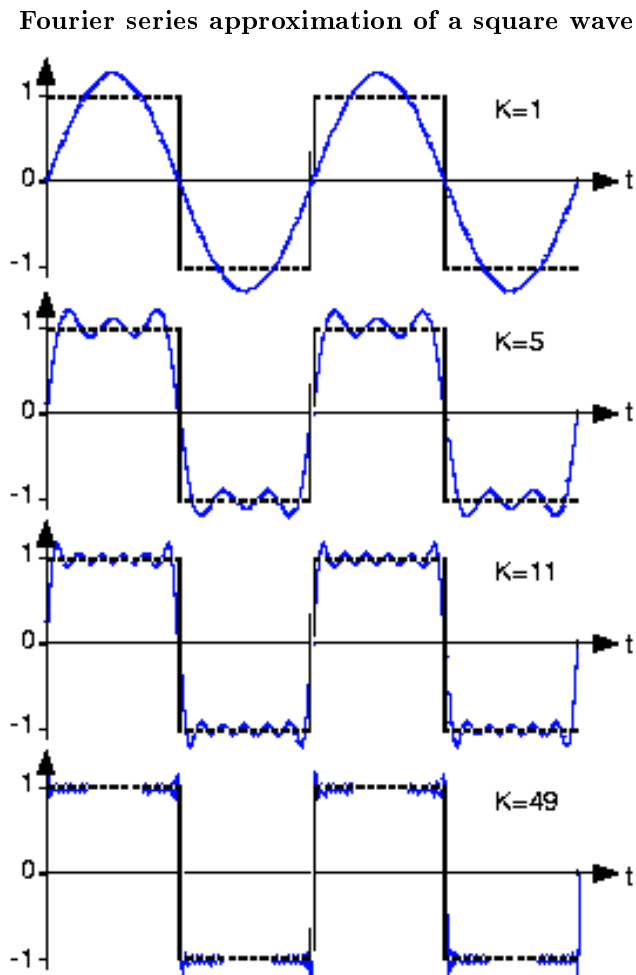


Figure 6.21: Fourier series approximation to $sq(t)$. The number of terms in the Fourier sum is indicated in each plot, and the square wave is shown as a dashed line over two periods.

When comparing the square wave to its Fourier series representation in Figure 6.21 (Fourier series approximation of a square wave), it is not clear that the two are equal. The fact that the square wave's Fourier series requires more terms for a given representation accuracy is not important. However, close inspection of Figure 6.21 (Fourier series approximation of a square wave) does reveal a potential issue: Does the Fourier series really equal the square wave at *all* values of t ? In particular, at each step-change in the square wave, the Fourier series exhibits a peak followed by rapid oscillations. As more terms are added to the series, the oscillations seem to become more rapid and smaller, but the peaks are not decreasing. Consider this mathematical question intuitively: Can a discontinuous function, like the square wave, be expressed as a sum, even an infinite one, of continuous ones? One should at least be suspicious, and in fact, it can't be thus expressed. This issue brought Fourier¹⁸ much criticism from the French Academy of Science (Laplace, Legendre, and Lagrange comprised the review committee) for several years after its presentation on 1807. It was not resolved for also a century, and its resolution is interesting and important to understand from a

¹⁸<http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Fourier.html>

practical viewpoint.

The extraneous peaks in the square wave's Fourier series *never* disappear; they are termed **Gibb's phenomenon** after the American physicist Josiah Willard Gibbs. They occur whenever the signal is discontinuous, and will always be present whenever the signal has jumps.

6.11.2.3 Redefine Equality

Let's return to the question of equality; how can the equal sign in the definition of the Fourier series be justified? The partial answer is that pointwise—each and every value of t —equality is *not* guaranteed. What mathematicians later in the nineteenth century showed was that the rms error of the Fourier series was always zero.

$$\lim_{K \rightarrow \infty} rms(\epsilon_K) = 0 \quad (6.52)$$

What this means is that the difference between an actual signal and its Fourier series representation may not be zero, but the square of this quantity has *zero* integral! It is through the eyes of the rms value that we define equality: Two signals $s_1(t)$, $s_2(t)$ are said to be equal in the **mean square** if $rms(s_1 - s_2) = 0$. These signals are said to be equal **pointwise** if $s_1(t) = s_2(t)$ for all values of t . For Fourier series, Gibb's phenomenon peaks have finite height and zero width: The error differs from zero only at isolated points—whenever the periodic signal contains discontinuities—and equals about 9% of the size of the discontinuity. The value of a function at a finite set of points does not affect its integral. This effect underlies the reason why defining the value of a discontinuous function at its discontinuity is meaningless. Whatever you pick for a value has no practical relevance for either the signal's spectrum or for how a system responds to the signal. The Fourier series value "at" the discontinuity is the average of the values on either side of the jump.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10092/latest/FFTSymbolic.llb>

6.12 Fourier Series Wrap-Up¹⁹

Below we will highlight some of the most important concepts about the Fourier Series (Section 6.2) and our understanding of it through eigenfunctions and eigenvalues. Hopefully you are familiar with all of this material, so this document will simply serve as a refresher, but if not, then refer to the many links below for more information on the various ideas and topics.

1. We can represent a periodic function (Section 6.1) (or a function on an interval) $f(t)$ as a combination of complex exponentials (Section 1.6):

$$f(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 n t}) \quad (6.53)$$

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-(j\omega_0 n t)} dt \quad (6.54)$$

Where the fourier coefficients, c_n , approximately equal how much of frequency $\omega_0 n$ is in the signal.

2. Since $e^{j\omega_0 n t}$ are eigenfunctions of LTI systems (Section 5.5), we can interpret the action of a system on a signal in terms of its eigenvalues (Section 5.2):

$$H(j\omega_0 n) = \int_{-\infty}^{\infty} h(t) e^{-(j\omega_0 n t)} dt \quad (6.55)$$

¹⁹This content is available online at <<http://cnx.org/content/m10749/2.4/>>.

- $|H(j\omega_0 n)|$ is large \Rightarrow system *accentuates* frequency $\omega_0 n$
 - $|H(j\omega_0 n)|$ is small \Rightarrow system *attenuates* frequency $\omega_0 n$
3. In addition, the $\{c_n\}$ of a periodic function $f(t)$ can tell us about:
 - symmetries in $f(t)$
 - smoothness of $f(t)$, where smoothness can be interpreted as the decay rate of $|c_n|$.
 4. We can *approximate* a function by re-synthesizing using only some of the Fourier coefficients (truncating the F.S.)

$$f_N'(t) = \sum_{n \leq |N|} (c_n e^{j\omega_0 n t}) \quad (6.56)$$

This approximation works well where $f(t)$ is continuous, but not so well where $f(t)$ is discontinuous. This idea is explained by Gibb's Phenomena.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10749/latest/TransferFunctions.llb>

Solutions to Exercises in Chapter 6

Solution to Exercise 6.1 (p. 113)

The tricky part of the problem is finding a way to represent the above function in terms of its basis, $e^{j\omega_0 t}$. To do this, we will use our knowledge of Euler's Relation (Section 1.6.2: Euler's Relation) to represent our cosine function in terms of the exponential.

$$f(t) = \frac{1}{2} \left(e^{j\omega_0 t} + e^{-(j\omega_0 t)} \right)$$

Now from this form of our function and from (6.3), by inspection we can see that our fourier coefficients will be:

$$c_n = \begin{cases} \frac{1}{2} & \text{if } |n| = 1 \\ 0 & \text{otherwise} \end{cases}$$

Solution to Exercise 6.2 (p. 113)

As done in the previous example, we will again use Euler's Relation (Section 1.6.2: Euler's Relation) to represent our sine function in terms of exponential functions.

$$f(t) = \frac{1}{2j} \left(e^{j\omega_0 t} - e^{-(j\omega_0 t)} \right)$$

And so our fourier coefficients are

$$c_n = \begin{cases} \frac{-j}{2} & \text{if } n = -1 \\ \frac{j}{2} & \text{if } n = 1 \\ 0 & \text{otherwise} \end{cases}$$

Solution to Exercise 6.3 (p. 113)

Once again we will use the same technique as was used in the previous two problems. The break down of our function yields

$$f(t) = 3 + 4 \left(\frac{1}{2} \right) \left(e^{j\omega_0 t} + e^{-(j\omega_0 t)} \right) + 2 \left(\frac{1}{2} \right) \left(e^{j2\omega_0 t} + e^{-(j2\omega_0 t)} \right)$$

And from this we can find our fourier coefficients to be:

$$c_n = \begin{cases} 3 & \text{if } n = 0 \\ 2 & \text{if } |n| = 1 \\ 1 & \text{if } |n| = 2 \\ 0 & \text{otherwise} \end{cases}$$

Solution to Exercise 6.4 (p. 114)

We will begin by plugging our above function, $f(t)$, into (6.4). Our interval of integration will now change to match the interval specified by the function.

$$c_n = \frac{1}{T} \int_{-T_1}^{T_1} (1) e^{-j\omega_0 n t} dt$$

Notice that we must consider two cases: $n = 0$ and $n \neq 0$. For $n = 0$ we can tell by inspection that we will get

$$c_n = \frac{2T_1}{T}, \quad n = 0$$

For $n \neq 0$, we will need to take a few more steps to solve. We can begin by looking at the basic integral of the exponential we have. Remembering our calculus, we are ready to integrate:

$$c_n = \frac{1}{T} \left(\frac{1}{j\omega_0 n} \right) e^{-(j\omega_0 n t)} \Big|_{t=-T_1}^{T_1}$$

Let us now evaluate the exponential functions for the given limits and expand our equation to:

$$c_n = \frac{1}{T} \left(\frac{1}{-j\omega_0 n} \right) \left(e^{-j\omega_0 n T_1} - e^{j\omega_0 n T_1} \right)$$

Now if we multiply the right side of our equation by $\frac{2j}{2j}$ and distribute our negative sign into the parenthesis, we can utilize Euler's Relation (Section 1.6.2: Euler's Relation) to greatly simplify our expression into:

$$c_n = \frac{1}{T} \left(\frac{2j}{j\omega_0 n} \right) \sin(\omega_0 n T_1)$$

Now, recall earlier that we defined $\omega_0 = \frac{2\pi}{T}$. We can solve this equation for T and substitute in.

$$c_n = \frac{2j\omega_0}{j\omega_0 n 2\pi} \sin(\omega_0 n T_1)$$

And finally, if we make a few simple cancellations we will arrive at our final answer for the Fourier coefficients of $f(t)$:

$$c_n = \frac{\sin(\omega_0 n T_1)}{n\pi}, \quad n \neq 0$$

Solution to Exercise 6.5 (p. 121)

$(|c_n|)^2 < \infty$ for $f(t)$ to have finite energy.

Solution to Exercise 6.6 (p. 121)

Yes, because $(|c_n|)^2 = \frac{1}{n^2}$, which is summable.

Solution to Exercise 6.7 (p. 121)

No, because $(|c_n|)^2 = \frac{1}{n}$, which is not summable.

Solution to Exercise 6.8 (p. 127)

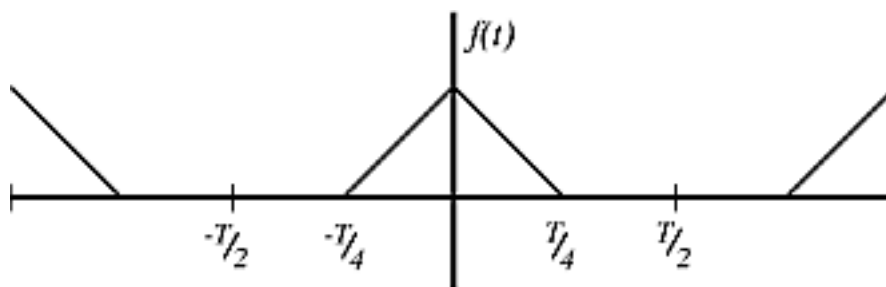


Figure 6.22: A triangle pulse train with a period of $\frac{T}{4}$.

Chapter 7

Discrete Fourier Transform

7.1 Fourier Analysis¹

Fourier analysis is fundamental to understanding the behavior of signals and systems. This is a result of the fact that sinusoids are Eigenfunctions (Section 5.5) of linear, time-invariant (LTI) (Section 2.1) systems. This is to say that if we pass any particular sinusoid through a LTI system, we get a scaled version of that same sinusoid on the output. Then, since Fourier analysis allows us to redefine the signals in terms of sinusoids, all we need to do is determine how any given system effects all possible sinusoids (its transfer function²) and we have a complete understanding of the system. Furthermore, since we are able to define the passage of sinusoids through a system as multiplication of that sinusoid by the transfer function at the same frequency, we can convert the passage of any signal through a system from convolution (Section 3.3) (in time) to multiplication (in frequency). These ideas are what give Fourier analysis its power.

Now, after hopefully having sold you on the value of this method of analysis, we must examine exactly what we mean by Fourier analysis. The four Fourier transforms that comprise this analysis are the Fourier Series³, Continuous-Time Fourier Transform (Section 11.1), Discrete-Time Fourier Transform (Section 10.4) and Discrete Fourier Transform⁴. For this document, we will view the Laplace Transform (Section 13.1) and Z-Transform (Section 14.2) as simply extensions of the CTFT and DTFT respectively. All of these transforms act essentially the same way, by converting a signal in time to an equivalent signal in frequency (sinusoids). However, depending on the nature of a specific signal *i.e.* whether it is finite- or infinite-length and whether it is discrete- or continuous-time) there is an appropriate transform to convert the signal into the frequency domain. Below is a table of the four Fourier transforms and when each is appropriate. It also includes the relevant convolution for the specified space.

¹This content is available online at <<http://cnx.org/content/m10096/2.10/>>.

²"Transfer Functions" <<http://cnx.org/content/m0028/latest/>>

³"Continuous-Time Fourier Series (CTFS)" <<http://cnx.org/content/m10097/latest/>>

⁴"Discrete Fourier Transform(DTFT)" <<http://cnx.org/content/m0502/latest/>>

Table of Fourier Representations

Transform	Time Domain	Frequency Domain	Convolution
Continuous-Time Fourier Series	$L^2([0, T])$	$l^2(\mathbb{Z})$	Continuous-Time Circular
Continuous-Time Fourier Transform	$L^2(\mathbb{R})$	$L^2(\mathbb{R})$	Continuous-Time Linear
Discrete-Time Fourier Transform	$l^2(\mathbb{Z})$	$L^2([0, 2\pi))$	Discrete-Time Linear
Discrete Fourier Transform	$l^2([0, N - 1])$	$l^2([0, N - 1])$	Discrete-Time Circular

7.2 Fourier Analysis in Complex Spaces⁵

7.2.1 Introduction

By now you should be familiar with the derivation of the Fourier series (Section 6.2) for continuous-time, periodic (Section 6.1) functions. This derivation leads us to the following equations that you should be quite familiar with:

$$f(t) = \sum_n (c_n e^{j\omega_0 n t}) \quad (7.1)$$

$$\begin{aligned} c_n &= \frac{1}{T} \int_n f(t) e^{-(j\omega_0 n t)} dt \\ &= \frac{1}{T} \langle f, e^{j\omega_0 n t} \rangle \end{aligned} \quad (7.2)$$

where c_n tells us the amount of frequency $\omega_0 n$ in $f(t)$.

In this module, we will derive a similar expansion for discrete-time, periodic functions. In doing so, we will derive the **Discrete Time Fourier Series** (DTFS), or the Discrete Fourier Transform (Section 10.2) (DFT).

7.2.2 Derivation of DTFS

Much like a periodic, continuous-time function can be thought of as a function on the interval $[0, T]$

⁵This content is available online at <http://cnx.org/content/m10784/2.7/>.

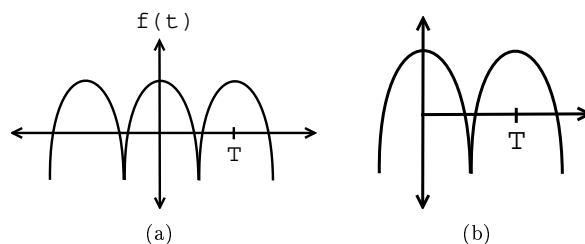


Figure 7.1: We will just consider one interval of the periodic function throughout this section. (a) Periodic Function (b) Function on the interval $[0, T]$

A periodic, discrete-time signal (with period N) can be thought of as a *finite* set of numbers. For example, say we have the following set of numbers that describe a periodic, discrete-time signal, where $N = 4$:

$$\{\dots, 3, 2, -2, 1, 3, \dots\}$$

We can represent this signal as either a periodic signal or as just a single interval as follows:

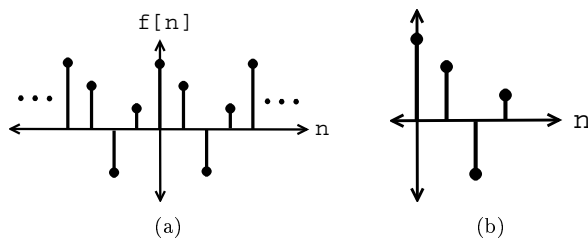


Figure 7.2: Here we can look at just one period of the signal that has a vector length of four and is contained in \mathbb{C}^4 . (a) Periodic Function (b) Function on the interval $[0, T]$

NOTE: The set of discrete time signals with period N equal \mathbb{C}^N .

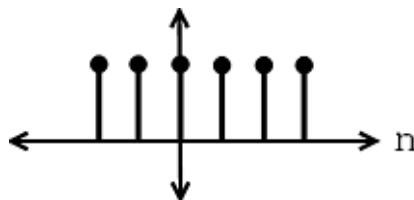
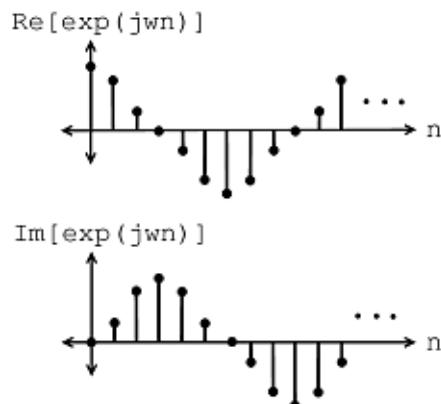
Just like the continuous case, we are going to form a basis using **harmonic sinusoids**. Before we look into this, it will be worth our time to look at the discrete-time, complex sinusoids in a little more detail.

7.2.2.1 Complex Sinusoids

If you are familiar with the basic sinusoid signal⁶ and with complex exponentials (Section 1.6) then you should not have any problem understanding this section. In most texts, you will see the the discrete-time, complex sinusoid noted as:

$$e^{j\omega n}$$

⁶"Elemental Signals" <<http://cnx.org/content/m0004/latest/>>

Example 7.1**Figure 7.3:** Complex sinusoid with frequency $\omega = 0$ **Example 7.2****Figure 7.4:** Complex sinusoid with frequency $\omega = \frac{\pi}{4}$ **7.2.2.1.1 In the Complex Plane**

The complex sinusoid can be directly mapped onto our complex plane⁷, which allows us to easily visualize changes to the complex sinusoid and extract certain properties. The absolute value of our complex sinusoid has the following characteristic:

$$|e^{j\omega n}| = 1, \quad n \in \mathbb{R} \quad (7.3)$$

which tells that our complex sinusoid only takes values on the unit circle. As for the angle, the following statement holds true:

$$\angle e^{j\omega n} = \omega n \quad (7.4)$$

As n increases, we can picture $e^{j\omega n}$ equaling the values we get moving counterclockwise around the unit circle. See Figure 7.5 for an illustration:

⁷"The Complex Plane" <<http://cnx.org/content/m10596/latest/>>

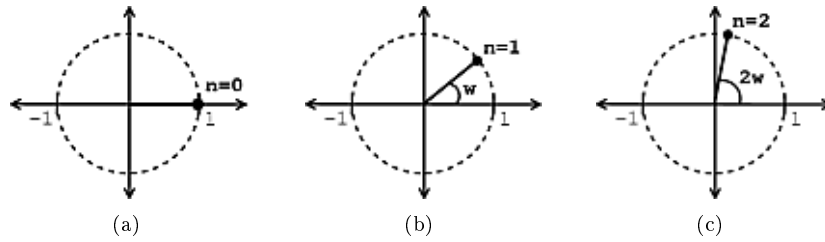


Figure 7.5: These images show that as n increases, the value of $e^{j\omega n}$ moves around the unit circle counterclockwise. (a) $n = 0$ (b) $n = 1$ (c) $n = 2$

NOTE: For $e^{j\omega n}$ to be periodic (Section 6.1), we need $e^{j\omega N} = 1$ for some N .

Example 7.3

For our first example let us look at a periodic signal where $\omega = \frac{2\pi}{7}$ and $N = 7$.

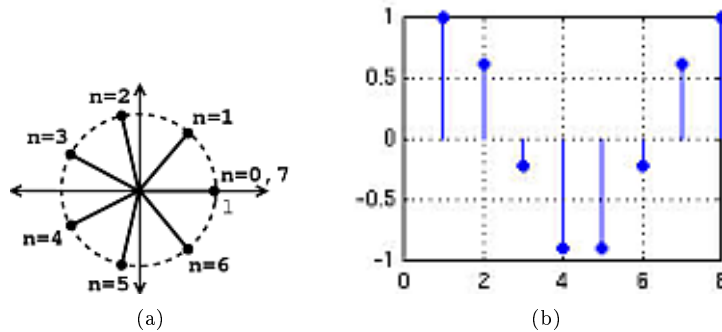


Figure 7.6: (a) $N = 7$ (b) Here we have a plot of $\text{Re}\left(e^{j\frac{2\pi}{7}n}\right)$.

Example 7.4

Now let us look at the results of plotting a non-periodic signal where $\omega = 1$ and $N = 7$.

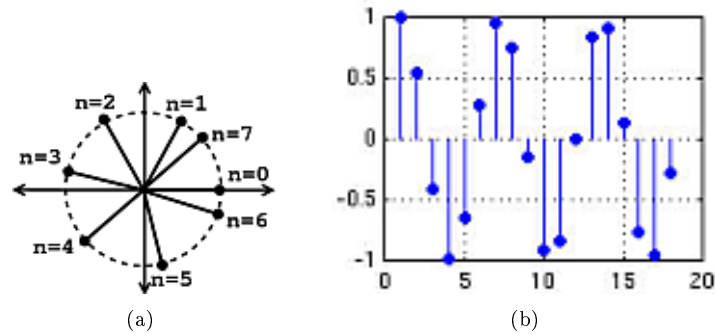


Figure 7.7: (a) $N = 7$ (b) Here we have a plot of $\text{Re}(e^{jn})$.

7.2.2.1.2 Aliasing

Our complex sinusoids have the following property:

$$e^{j\omega n} = e^{j(\omega+2\pi)n} \quad (7.5)$$

Given this property, if we have a sinusoid with frequency $\omega + 2\pi$, then this signal "aliases" to a sinusoid with frequency ω .

NOTE: Each $e^{j\omega n}$ is unique for $\omega \in [0, 2\pi)$

7.2.2.1.3 "Negative" Frequencies

If we are given a signal with frequency $\pi < \omega < 2\pi$, then this signal will be represented on our complex plane as:

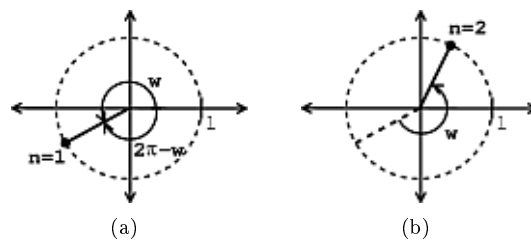


Figure 7.8: Plot of our complex sinusoid with a frequency greater than π .

From the above images, the value of our complex sinusoid on the complex plane may be more easily interpreted as cycling "backwards" (clockwise) around the unit circle with frequency $2\pi - \omega$. Rotating counterclockwise by w is the same as rotating clockwise by $2\pi - \omega$.

Example 7.5

Let us plot our complex sinusoid, $e^{j\omega n}$, where we have $\omega = \frac{5\pi}{4}$ and $n = 1$.

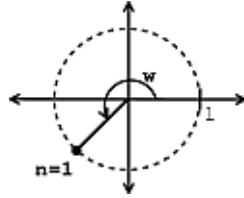


Figure 7.9: The above plot of our given frequency is identical to that of one where $\omega = -\left(\frac{3\pi}{4}\right)$.

This plot is the same as a sinusoid with "negative" frequency $-\left(\frac{3\pi}{4}\right)$.

POINT: It makes more physical sense to chose $[-\pi, \pi)$ as the interval for ω .

Remember that $e^{j\omega n}$ and $e^{-(j\omega n)}$ are **conjugates**. This gives us the following notation and property:

$$e^{j\omega n^*} = e^{-(j\omega n)} \tag{7.6}$$

The real parts of both exponentials in the above equation are the same; the imaginary parts are negative of one another. This idea is the basic definition of a conjugate.

Now that we have looked over the concepts of complex sinusoids, let us turn our attention back to finding a basis for discrete-time, periodic signals. After looking at all the complex sinusoids, we must answer the question of which discrete-time sinusoids do we need to represent periodic sequences with a period N .

EQUIVALENT QUESTION: Find a set of vectors $b_k = e^{j\omega_k n}$, $n = \{0, \dots, N - 1\}$ such that $\{b_k\}$ are a **basis** for \mathbb{C}^n

In answer to the above question, let us try the "harmonic" sinusoids with a fundamental frequency $\omega_0 = \frac{2\pi}{N}$:

Harmonic Sinusoid

$$e^{j\frac{2\pi}{N}kn} \tag{7.7}$$

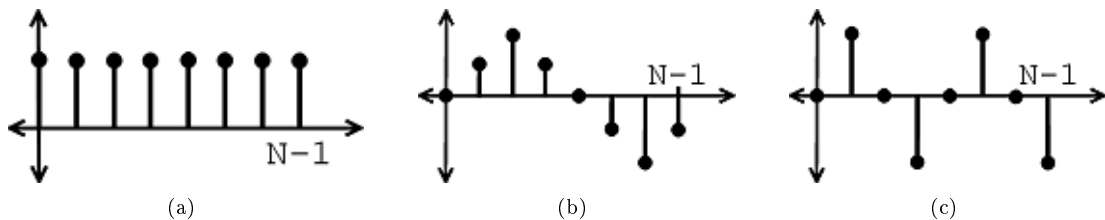


Figure 7.10: Examples of our Harmonic Sinusoids (a) Harmonic sinusoid with $k = 0$ (b) Imaginary part of sinusoid, $Im\left(e^{j\frac{2\pi}{N}1n}\right)$, with $k = 1$ (c) Imaginary part of sinusoid, $Im\left(e^{j\frac{2\pi}{N}2n}\right)$, with $k = 2$

$e^{j\frac{2\pi}{N}kn}$ is periodic with period N and has k "cycles" between $n = 0$ and $n = N - 1$.

Theorem 7.1:

If we let

$$b_k[n] = \frac{1}{\sqrt{N}} e^{j\frac{2\pi}{N}kn} \quad , \quad n = \{0, \dots, N-1\}$$

where the exponential term is a vector in \mathbb{C}^N , then $\{b_k\}_{k=\{0, \dots, N-1\}}$ is an **orthonormal basis** (Section 15.7.3: Orthonormal Basis) for \mathbb{C}^N .

Proof: First of all, we must show $\{b_k\}$ is orthonormal, i.e. $\langle b_k, b_l \rangle = \delta_{kl}$

$$\begin{aligned} \langle b_k, b_l \rangle &= \sum_{n=0}^{N-1} (b_k[n] b_l[n]^*) = \frac{1}{N} \sum_{n=0}^{N-1} \left(e^{j\frac{2\pi}{N}kn} e^{-j\frac{2\pi}{N}ln} \right) \\ \langle b_k, b_l \rangle &= \frac{1}{N} \sum_{n=0}^{N-1} \left(e^{j\frac{2\pi}{N}(l-k)n} \right) \end{aligned} \quad (7.8)$$

If $l = k$, then

$$\begin{aligned} \langle b_k, b_l \rangle &= \frac{1}{N} \sum_{n=0}^{N-1} (1) \\ &= 1 \end{aligned} \quad (7.9)$$

If $l \neq k$, then we must use the "partial summation formula" shown below:

$$\begin{aligned} \sum_{n=0}^{N-1} (\alpha^n) &= \sum_{n=0}^{\infty} (\alpha^n) - \sum_{n=N}^{\infty} (\alpha^n) = \frac{1}{1-\alpha} - \frac{\alpha^N}{1-\alpha} = \frac{1-\alpha^N}{1-\alpha} \\ \langle b_k, b_l \rangle &= \frac{1}{N} \sum_{n=0}^{N-1} \left(e^{j\frac{2\pi}{N}(l-k)n} \right) \end{aligned}$$

where in the above equation we can say that $\alpha = e^{j\frac{2\pi}{N}(l-k)}$, and thus we can see how this is in the form needed to utilize our partial summation formula.

$$\langle b_k, b_l \rangle = \frac{1}{N} \left(\frac{1 - e^{j\frac{2\pi}{N}(l-k)N}}{1 - e^{j\frac{2\pi}{N}(l-k)}} \right) = \frac{1}{N} \left(\frac{1 - 1}{1 - e^{j\frac{2\pi}{N}(l-k)}} \right) = 0$$

So,

$$\langle b_k, b_l \rangle = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases} \quad (7.10)$$

Therefore: $\{b_k\}$ is an orthonormal set. $\{b_k\}$ is also a basis (Section 5.1.3: Basis), since there are N vectors which are linearly independent (Section 5.1.1: Linear Independence) (orthogonality implies linear independence).

And finally, we have shown that the harmonic sinusoids $\left\{ \frac{1}{\sqrt{N}} e^{j\frac{2\pi}{N}kn} \right\}$ form an orthonormal basis for \mathbb{C}^n

7.2.2.2 Discrete-Time Fourier Series (DTFS)

Using the steps shown above in the derivation and our previous understanding of Hilbert Spaces (Section 15.3) and Orthogonal Expansions (Section 15.8), the rest of the derivation is automatic. Given a discrete-time, periodic signal (vector in \mathbb{C}^n) $f[n]$, we can write:

$$f[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(c_k e^{j \frac{2\pi}{N} kn} \right) \quad (7.11)$$

$$c_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \left(f[n] e^{-j \frac{2\pi}{N} kn} \right) \quad (7.12)$$

Note: Most people collect both the $\frac{1}{\sqrt{N}}$ terms into the expression for c_k .

DISCRETE TIME FOURIER SERIES: Here is the common form of the DTFS with the above note taken into account:

$$f[n] = \sum_{k=0}^{N-1} \left(c_k e^{j \frac{2\pi}{N} kn} \right)$$

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} \left(f[n] e^{-j \frac{2\pi}{N} kn} \right)$$

This what the `fft` command in MATLAB does.

7.3 Matrix Equation for the DTFS⁸

The DTFS (Section 7.2.2.2: Discrete-Time Fourier Series (DTFS)) is just a change of basis (Section 5.1.3: Basis) in \mathbb{C}^N . To start, we have $f[n]$ in terms of the *standard basis*.

$$\begin{aligned} f[n] &= f[0]e_0 + f[1]e_1 + \cdots + f[N-1]e_{N-1} \\ &= \sum_{k=0}^{N-1} (f[k] \delta[k-n]) \end{aligned} \quad (7.13)$$

$$\begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ \vdots \\ f[N-1] \end{pmatrix} = \begin{pmatrix} f[0] \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ f[1] \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ f[2] \\ \vdots \\ 0 \end{pmatrix} + \cdots + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ f[N-1] \end{pmatrix} \quad (7.14)$$

Taking the DTFS, we can write $f[n]$ in terms of the sinusoidal Fourier basis

$$f[n] = \sum_{k=0}^{N-1} \left(c_k e^{j \frac{2\pi}{N} kn} \right) \quad (7.15)$$

$$\begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ \vdots \\ f[N-1] \end{pmatrix} = c_0 \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + c_1 \begin{pmatrix} 1 \\ e^{j \frac{2\pi}{N}} \\ e^{j \frac{4\pi}{N}} \\ \vdots \\ e^{j \frac{2\pi}{N}(N-1)} \end{pmatrix} + c_2 \begin{pmatrix} 1 \\ e^{j \frac{4\pi}{N}} \\ e^{j \frac{8\pi}{N}} \\ \vdots \\ e^{j \frac{4\pi}{N}(N-1)} \end{pmatrix} + \cdots \quad (7.16)$$

⁸This content is available online at <<http://cnx.org/content/m10771/2.6/>>.

We can form the basis matrix (we'll call it W here instead of B) by stacking the basis vectors in as columns

$$\begin{aligned}
 W &= \begin{pmatrix} b_0[n] & b_1[n] & \dots & b_{N-1}[n] \end{pmatrix} \\
 &= \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{j\frac{2\pi}{N}} & e^{j\frac{4\pi}{N}} & \dots & e^{j\frac{2\pi}{N}(N-1)} \\ 1 & e^{j\frac{4\pi}{N}} & e^{j\frac{8\pi}{N}} & \dots & e^{j\frac{2\pi}{N}2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & e^{j\frac{2\pi}{N}(N-1)} & e^{j\frac{2\pi}{N}2(N-1)} & \dots & e^{j\frac{2\pi}{N}(N-1)(N-1)} \end{pmatrix} \quad (7.17)
 \end{aligned}$$

with $b_k[n] = e^{j\frac{2\pi}{N}kn}$

NOTE: the entry in the k -th row and n -th column is $W_{j,k} = e^{j\frac{2\pi}{N}kn} = W_{n,k}$

So, here we have an additional symmetry

$$W = W^T \Rightarrow W^{T*} = W^* = \frac{1}{N}W^{-1}$$

(since $\{b_k[n]\}$ are orthogonal)

We can now rewrite the DTFS equations in matrix form where we have:

- \mathbf{f} = signal (vector in \mathbb{C}^N)
- \mathbf{c} = DTFS coeffs. (vector in \mathbb{C}^N)

"synthesis"	$\mathbf{f} = W\mathbf{c}$	$f[n] = \langle \mathbf{c}, b_n^* \rangle$
"analysis"	$\mathbf{c} = W^{T*}\mathbf{f} = W^*\mathbf{f}$	$c[k] = \langle \mathbf{f}, b_k \rangle$

Finding (and inverting) the DTFS is just *matrix multiplication*.

Everything in \mathbb{C}^N is *clean*: no limits, no convergence questions, just good ole matrix arithmetic.

7.4 Periodic Extension to DTFS⁹

7.4.1 Introduction

Now that we have an understanding of the discrete-time Fourier series (DTFS) (Section 7.2.2.2: Discrete-Time Fourier Series (DTFS)), we can consider the *periodic extension* of $c[k]$ (the Discrete-time Fourier coefficients). Figure 7.11 shows a simple illustration of how we can represent a sequence as a periodic signal mapped over an infinite number of intervals.

⁹This content is available online at <http://cnx.org/content/m10778/2.8/>.

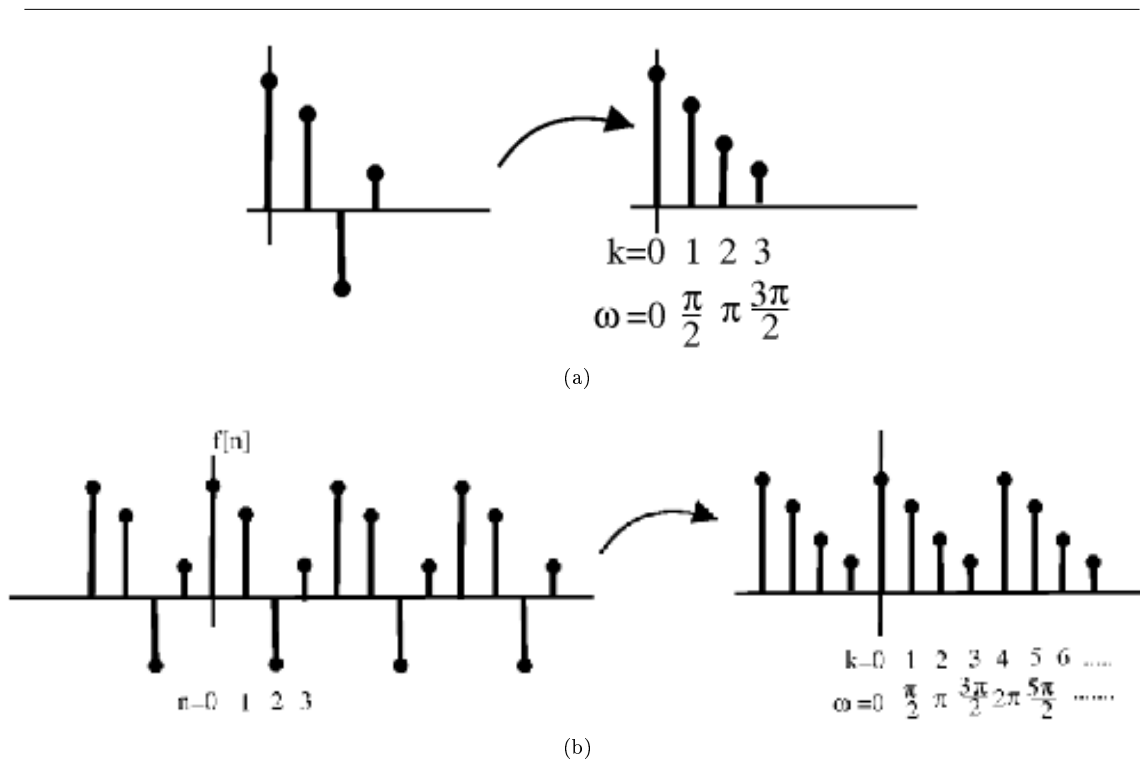


Figure 7.11: (a) vectors (b) periodic sequences

Exercise 7.1

(Solution on p. 163.)

Why does a periodic (Section 6.1) extension to the DTFS coefficients $c[k]$ make sense?

7.4.2 Examples

Example 7.6: Discrete time square wave

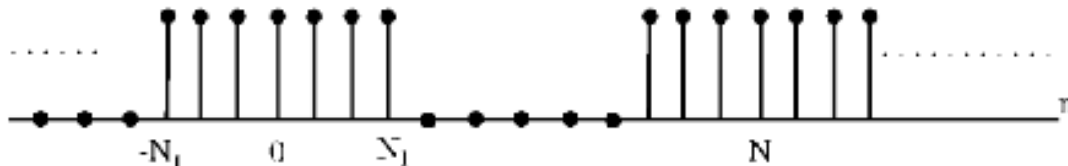


Figure 7.12

Calculate the DTFS $c[k]$ using:

$$c[k] = \frac{1}{N} \sum_{n=0}^{N-1} \left(f[n] e^{-j\frac{2\pi}{N}kn} \right) \quad (7.18)$$

Just like continuous time Fourier series, we can take the summation over any interval, so we have

$$c_k = \frac{1}{N} \sum_{n=-N_1}^{N_1} \left(e^{-j\frac{2\pi}{N}kn} \right) \quad (7.19)$$

Let $m = n + N_1$ (so we can get a geometric series starting at 0)

$$\begin{aligned} c_k &= \frac{1}{N} \sum_{m=0}^{2N_1} \left(e^{-j\frac{2\pi}{N}(m-N_1)k} \right) \\ &= \frac{1}{N} e^{j\frac{2\pi}{N}N_1k} \sum_{m=0}^{2N_1} \left(e^{-j\frac{2\pi}{N}mk} \right) \end{aligned} \quad (7.20)$$

Now, using the "partial summation formula"

$$\sum_{n=0}^M (a^n) = \frac{1 - a^{M+1}}{1 - a} \quad (7.21)$$

$$\begin{aligned} c_k &= \frac{1}{N} e^{j\frac{2\pi}{N}N_1k} \sum_{m=0}^{2N_1} \left(\left(e^{-j\frac{2\pi}{N}k} \right)^m \right) \\ &= \frac{1}{N} e^{j\frac{2\pi}{N}N_1k} \frac{1 - e^{-j\frac{2\pi}{N}(2N_1+1)k}}{1 - e^{-j\frac{2\pi}{N}k}} \end{aligned} \quad (7.22)$$

Manipulate to make this look like a sinc function (distribute):

$$\begin{aligned} c_k &= \frac{1}{N} \frac{e^{-j\frac{2\pi}{N}k} \left(e^{j\frac{2\pi}{N}k(N_1+\frac{1}{2})} - e^{-j\frac{2\pi}{N}k(N_1+\frac{1}{2})} \right)}{e^{-j\frac{2\pi}{N}k} \left(e^{j\frac{2\pi}{N}k\frac{1}{2}} - e^{-j\frac{2\pi}{N}k\frac{1}{2}} \right)} \\ &= \frac{1}{N} \frac{\sin\left(\frac{2\pi k(N_1+\frac{1}{2})}{N}\right)}{\sin\left(\frac{\pi k}{N}\right)} \\ &= \text{digital sinc} \end{aligned} \quad (7.23)$$

NOTE: It's periodic! Figure 7.13, Figure 7.14, and Figure 7.15 show our above function and coefficients for various values of N_1 .

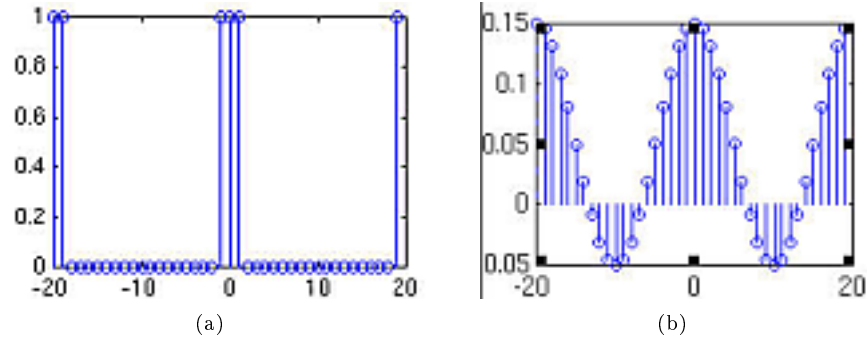


Figure 7.13: $N_1 = 1$ (a) Plot of $f[n]$. (b) Plot of $c[k]$.

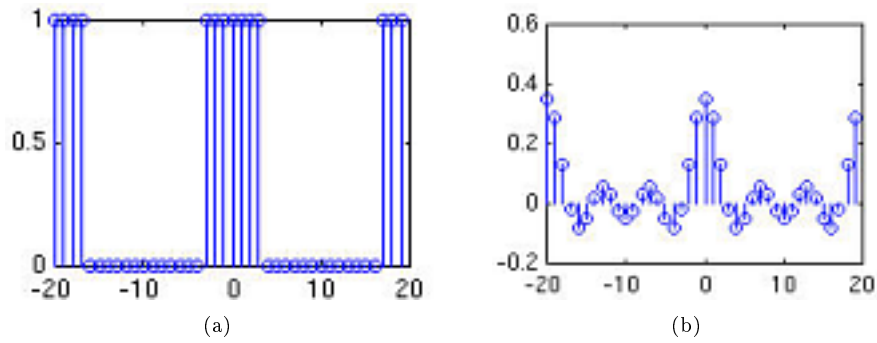


Figure 7.14: $N_1 = 3$ (a) Plot of $f[n]$. (b) Plot of $c[k]$.

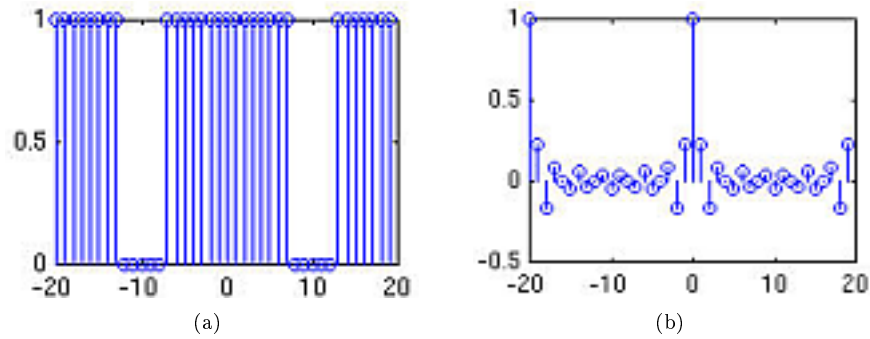


Figure 7.15: $N_1 = 7$ (a) Plot of $f[n]$. (b) Plot of $c[k]$.

7.5 Circular Shifts¹⁰

The many properties of the DFT (Section 7.2.2.2: Discrete-Time Fourier Series (DTFS)) become really straightforward (very similar to the Fourier Series (Section 6.2)) once we have once concept down: **Circular Shifts**.

7.5.1 Circular shifts

We can picture periodic (Section 6.1) sequences as having discrete points on a circle as the domain

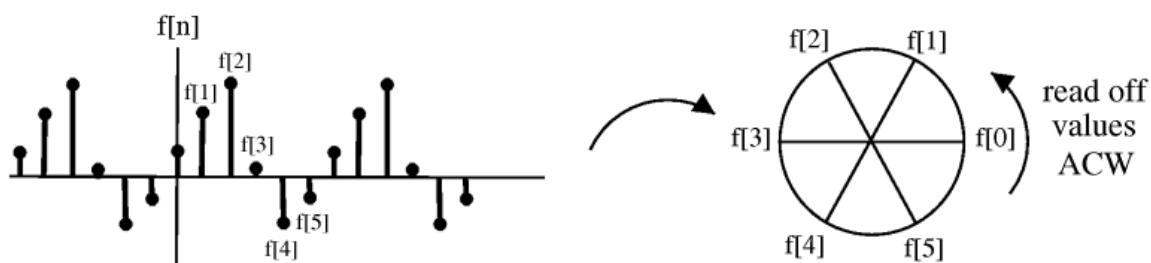


Figure 7.16

Shifting by m , $f(n+m)$, corresponds to rotating the cylinder m notches ACW (counter clockwise). For $m = -2$, we get a shift equal to that in the following illustration:

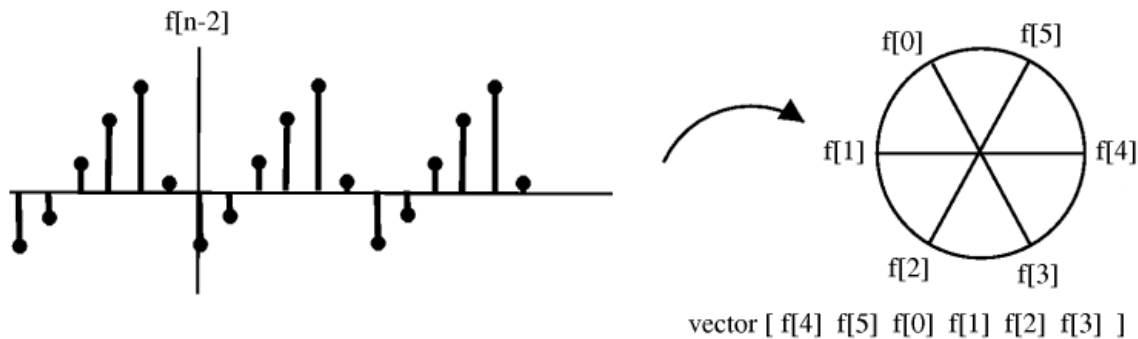


Figure 7.17: for $m = -2$

¹⁰This content is available online at <<http://cnx.org/content/m10780/2.6/>>.

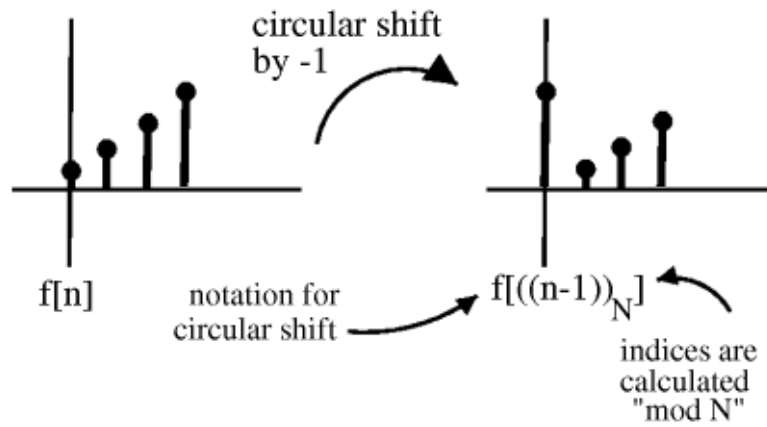
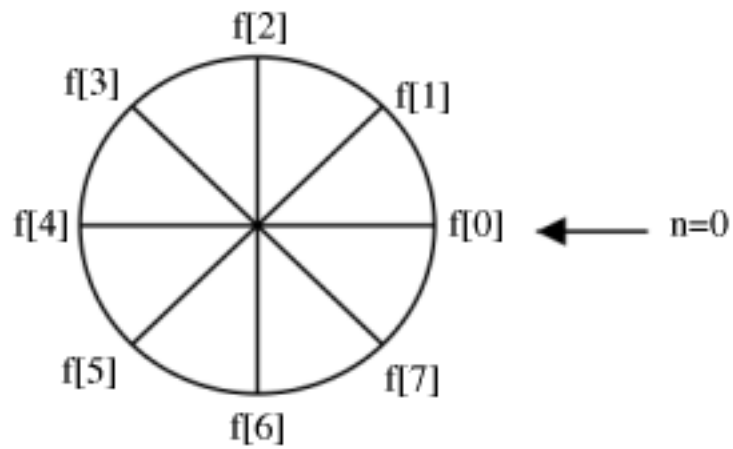


Figure 7.18

To cyclic shift we follow these steps:

- 1) Write $f(n)$ on a cylinder, ACW

Figure 7.19: $N = 8$

2) To cyclic shift by m , spin cylinder m spots ACW

$$(f[n] \rightarrow f[(n+m)_N])$$

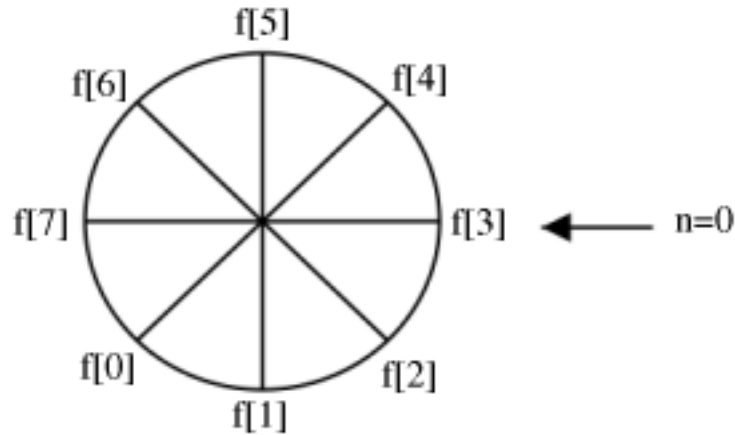


Figure 7.20: $m = -3$

Example 7.7

If $f(n) = [0, 1, 2, 3, 4, 5, 6, 7]$, then $f((n-3)_N) = [3, 4, 5, 6, 7, 0, 1, 2]$

It's called **circular shifting**, since we're moving around the circle. The usual shifting is called "linear shifting" (shifting along a line).

7.5.1.1 Notes on circular shifting

$$f[((n+N))_N] = f[n]$$

Spinning N spots is the same as spinning all the way around, or not spinning at all.

$$f[((n+N))_N] = f[(n-(N-m))_N]$$

Shifting ACW m is equivalent to shifting CW $N-m$

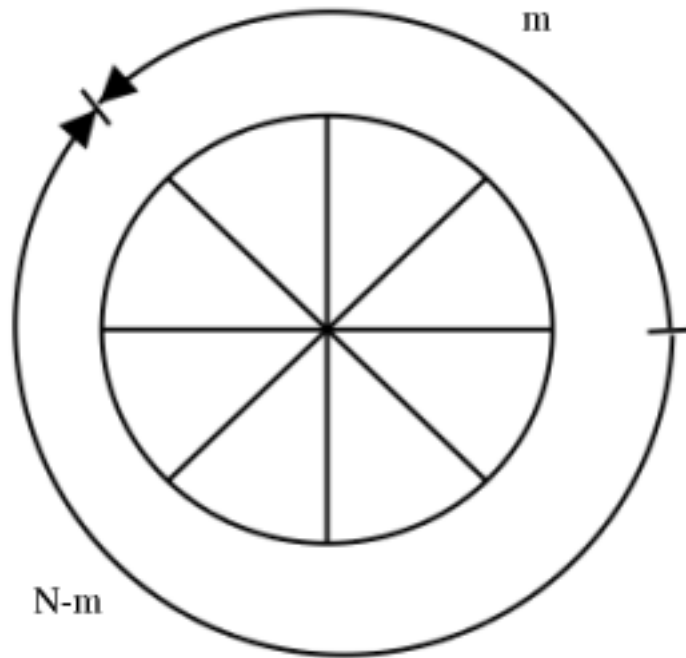


Figure 7.21

$$f [((-n))_N]$$

The above expression, simply writes the values of $f [n]$ clockwise.

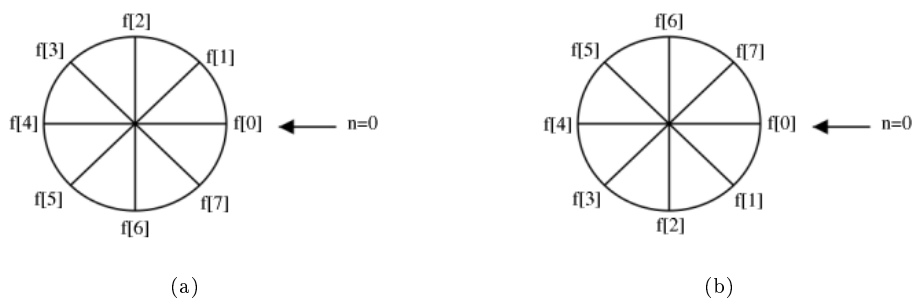


Figure 7.22: (a) $f[n]$ (b) $f[((-n))_N]$

7.5.2 Circular shifts and the DFT

Theorem 7.2: Circular Shifts and DFT
If

$$f[n] \stackrel{\text{DFT}}{\leftrightarrow} F[k]$$

then

$$f[((n-m))_N] \stackrel{\text{DFT}}{\leftrightarrow} e^{-j\frac{2\pi}{N}km} F[k]$$

(i.e. circular shift in time domain = phase shift in DFT)

Proof:

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} \left(F[k] e^{j\frac{2\pi}{N}kn} \right) \quad (7.24)$$

so phase shifting the DFT

$$\begin{aligned} f[n] &= \frac{1}{N} \sum_{k=0}^{N-1} \left(F[k] e^{-j\frac{2\pi}{N}kn} e^{j\frac{2\pi}{N}kn} \right) \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \left(F[k] e^{j\frac{2\pi}{N}k(n-m)} \right) \\ &= f[((n-m))_N] \end{aligned} \quad (7.25)$$

7.6 Circular Convolution and the DFT¹¹

7.6.1 Introduction

You should be familiar with Discrete-Time Convolution (Section 4.2), which tells us that given two discrete-time signals $x[n]$, the system's input, and $h[n]$, the system's response, we define the output of the system as

$$\begin{aligned} y[n] &= x[n] * h[n] \\ &= \sum_{k=-\infty}^{\infty} (x[k] h[n-k]) \end{aligned} \quad (7.26)$$

¹¹This content is available online at <<http://cnx.org/content/m10786/2.8/>>.

When we are given two DFTs (finite-length sequences usually of length N), we cannot just multiply them together as we do in the above convolution formula, often referred to as **linear convolution**. Because the DFTs are periodic, they have nonzero values for $n \geq N$ and thus the multiplication of these two DFTs will be nonzero for $n \geq N$. We need to define a new type of convolution operation that will result in our convolved signal being zero outside of the range $n = \{0, 1, \dots, N - 1\}$. This idea led to the development of **circular convolution**, also called cyclic or periodic convolution.

7.6.2 Circular Convolution Formula

What happens when we multiply two DFT's together, where $Y[k]$ is the DFT of $y[n]$?

$$Y[k] = F[k] H[k] \quad (7.27)$$

when $0 \leq k \leq N - 1$

Using the DFT synthesis formula for $y[n]$

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} \left(F[k] H[k] e^{j \frac{2\pi}{N} kn} \right) \quad (7.28)$$

And then applying the analysis formula $F[k] = \sum_{m=0}^{N-1} \left(f[m] e^{(-j) \frac{2\pi}{N} km} \right)$

$$\begin{aligned} y[n] &= \frac{1}{N} \sum_{k=0}^{N-1} \left(\left(\sum_{m=0}^{N-1} \left(f[m] e^{(-j) \frac{2\pi}{N} km} \right) \right) H[k] e^{j \frac{2\pi}{N} kn} \right) \\ &= \sum_{m=0}^{N-1} \left(f[m] \left(\frac{1}{N} \sum_{k=0}^{N-1} \left(H[k] e^{j \frac{2\pi}{N} k(n-m)} \right) \right) \right) \end{aligned} \quad (7.29)$$

where we can reduce the second summation found in the above equation into $h[((n-m))_N] = \frac{1}{N} \sum_{k=0}^{N-1} \left(H[k] e^{j \frac{2\pi}{N} k(n-m)} \right)$

$$y[n] = \sum_{m=0}^{N-1} \left(f[m] h[((n-m))_N] \right)$$

which equals circular convolution! When we have $0 \leq n \leq N - 1$ in the above, then we get:

$$y[n] \equiv (f[n] \circledast h[n]) \quad (7.30)$$

NOTE: The notation \circledast represents cyclic convolution "mod N ".

7.6.2.1 Steps for Cyclic Convolution

Steps for cyclic convolution are the same as the usual convo, except all index calculations are done "mod N " = "on the wheel"

Steps for Cyclic Convolution

- Step 1: "Plot" $f[m]$ and $h[((-m))_N]$

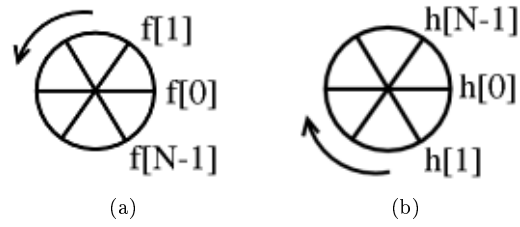


Figure 7.23: Step 1

- Step 2: "Spin" $h[(n-m)_N]$ n notches ACW (counter-clockwise) to get $h[((n-m)_N)]$ (i.e. Simply rotate the sequence, $h[n]$, clockwise by n steps).

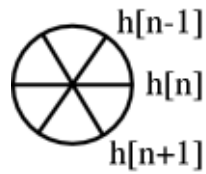


Figure 7.24: Step 2

- Step 3: Pointwise multiply the $f[m]$ wheel and the $h[(n-m)_N]$ wheel.

$$sum = y[n]$$

- Step 4: Repeat for all $0 \leq n \leq N - 1$

Example 7.8: Convolve ($n = 4$)

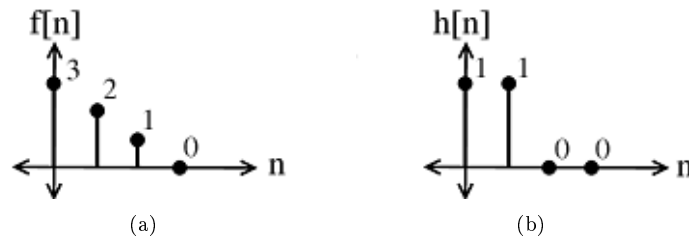


Figure 7.25: Two discrete-time signals to be convolved.

- $h[(-m)_N]$

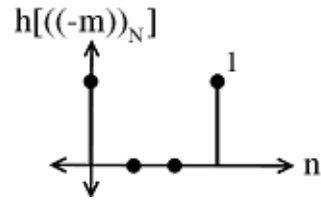


Figure 7.26

Multiply $f[m]$ and sum to yield: $y[0] = 3$

- $h[(1-m)_N]$

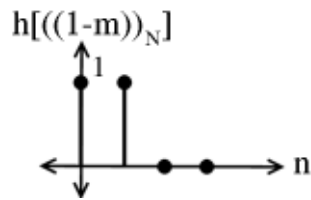


Figure 7.27

Multiply $f[m]$ and sum to yield: $y[1] = 5$

- $h[(2-m)_N]$

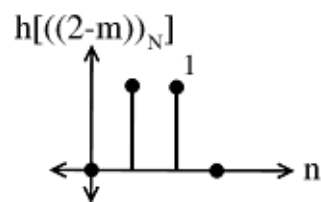


Figure 7.28

Multiply $f[m]$ and sum to yield: $y[2] = 3$

- $h[(3-m)_N]$

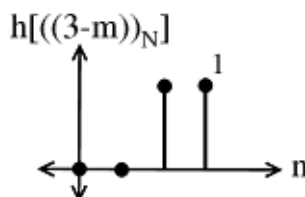


Figure 7.29

Multiply $f[m]$ and sum to yield: $y[3] = 1$

Example 7.9

The following demonstration allows you to explore this algorithm for circular convolution. See here¹² for instructions on how to use the demo.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10786/latest/DTCircularConvolution.llb>

7.6.2.2 Alternative Algorithm

Alternative Circular Convolution Algorithm

- Step 1: Calculate the DFT of $f[n]$ which yields $F[k]$ and calculate the DFT of $h[n]$ which yields $H[k]$.
- Step 2: Pointwise multiply $Y[k] = F[k] H[k]$
- Step 3: Inverse DFT $Y[k]$ which yields $y[n]$

Seems like a roundabout way of doing things, *but* it turns out that there are *extremely* fast ways to calculate the DFT of a sequence.

To circularly convolve 2 N -point sequences:

$$y[n] = \sum_{m=0}^{N-1} (f[m] h[((n-m))_N])$$

For each n : N multiples, $N-1$ additions

N points implies N^2 multiplications, $N(N-1)$ additions implies $O(N^2)$ complexity.

¹²"How to use the LabVIEW demos" <<http://cnx.org/content/m11550/latest/>>

Solutions to Exercises in Chapter 7

Solution to Exercise 7.1 (p. 151)

Aliasing: $b_k = e^{j\frac{2\pi}{N}kn}$

$$\begin{aligned} b_{k+N} &= e^{j\frac{2\pi}{N}(k+N)n} \\ &= e^{j\frac{2\pi}{N}kn} e^{j2\pi n} \\ &= e^{j\frac{2\pi}{N}kn} \\ &= b_k \end{aligned} \tag{7.31}$$

→ DTFS coefficients are also periodic with period N .

Chapter 8

Fast Fourier Transform (FFT)

8.1 DFT: Fast Fourier Transform¹

We now have a way of computing the spectrum for an arbitrary signal: The Discrete Fourier Transform (DFT)² computes the spectrum at N equally spaced frequencies from a length- N sequence. An issue that never arises in analog "computation," like that performed by a circuit, is how much work it takes to perform the signal processing operation such as filtering. In computation, this consideration translates to the number of basic computational steps required to perform the needed processing. The number of steps, known as the **complexity**, becomes equivalent to how long the computation takes (how long must we wait for an answer). Complexity is not so much tied to specific computers or programming languages but to how many steps are required on any computer. Thus, a procedure's stated complexity says that the time taken will be **proportional** to some function of the amount of data used in the computation and the amount demanded.

For example, consider the formula for the discrete Fourier transform. For each frequency we chose, we must multiply each signal value by a complex number and add together the results. For a real-valued signal, each real-times-complex multiplication requires two real multiplications, meaning we have $2N$ multiplications to perform. To add the results together, we must keep the real and imaginary parts separate. Adding N numbers requires $N - 1$ additions. Consequently, each frequency requires $2N + 2(N - 1) = 4N - 2$ basic computational steps. As we have N frequencies, the total number of computations is $N(4N - 2)$.

In complexity calculations, we only worry about what happens as the data lengths increase, and take the dominant term—here the $4N^2$ term—as reflecting how much work is involved in making the computation. As multiplicative constants don't matter since we are making a "proportional to" evaluation, we find the DFT is an $O(N^2)$ computational procedure. This notation is read "order N -squared". Thus, if we double the length of the data, we would expect that the computation time to approximately quadruple.

Exercise 8.1

(Solution on p. 170.)

In making the complexity evaluation for the DFT, we assumed the data to be real. Three questions emerge. First of all, the spectra of such signals have conjugate symmetry, meaning that negative frequency components ($k = [\frac{N}{2} + 1, \dots, N + 1]$ in the DFT³) can be computed from the corresponding positive frequency components. Does this symmetry change the DFT's complexity?

Secondly, suppose the data are complex-valued; what is the DFT's complexity now?

Finally, a less important but interesting question is suppose we want K frequency values instead of N ; now what is the complexity?

¹This content is available online at <<http://cnx.org/content/m0504/2.8/>>.

²"Discrete Fourier Transform(DTFT)", (1) : Discrete Fourier transform <<http://cnx.org/content/m0502/latest/#eqn1>>

³"Discrete Fourier Transform(DTFT)", (1) : Discrete Fourier transform <<http://cnx.org/content/m0502/latest/#eqn1>>

8.2 The Fast Fourier Transform (FFT)⁴

8.2.1 Introduction

The Fast Fourier Transform (FFT) is an efficient $O(N\log N)$ algorithm for calculating DFTs

- originally discovered by Gauss in the early 1800's
- rediscovered by Cooley and Tukey at IBM in the 1960's
- C.S. Burrus, Rice University's very own Dean of Engineering, literally "wrote the book" on fast DFT algorithms.

The FFT⁵ exploits symmetries in the W matrix to take a "divide and conquer" approach. We won't talk about the actual FFT algorithm here, see these notes⁶ if you are interested in reading a little more on the idea behind FFT.

8.2.2 Speed Comparison

How much better is $O(N\log N)$ than $O(N^2)$?

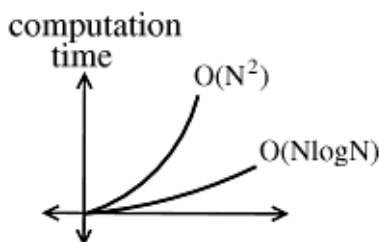


Figure 8.1: This figure shows how much slower the computation time of an $O(N\log N)$ process grows.

N	10	100	1000	10^6	10^9
N^2	100	10^4	10^6	10^{12}	10^{18}
$N\log N$	1	200	3000	6×10^6	9×10^9

Say you have a 1 MFLOP machine (a million "floating point" operations per second). Let $N = 1\text{million} = 10^6$.

An $O(N^2)$ algorithm takes 10^{12} flops $\rightarrow 10^6$ seconds $\simeq 11.5$ days.

An $O(N\log N)$ algorithm takes 6×10^6 Flops $\rightarrow 6$ seconds.

NOTE: $N = 1\text{million}$ is **not** unreasonable.

Example 8.1

3 megapixel digital camera spits out 3×10^6 numbers for each picture. So for two N point sequences $f[n]$ and $h[n]$. If computing $(f[n] \otimes h[n])$ directly: $O(N^2)$ operations.

taking FFTs – $O(N\log N)$
 multiplying FFTs – $O(N)$

⁴This content is available online at <http://cnx.org/content/m10783/2.5/>.

⁵"Fast Fourier Transform (FFT)" <http://cnx.org/content/m10250/latest/>

⁶"Fast Fourier Transform (FFT)" <http://cnx.org/content/m10250/latest/>

inverse FFTs – $O(N \log N)$.
 the total complexity is $O(N \log N)$.

NOTE: FFT + digital computer were almost completely responsible for the "explosion" of DSP in the 60's.

NOTE: Rice was (and still is) one of the places to do research in DSP.

8.3 Deriving the Fast Fourier Transform⁷

To derive the FFT, we assume that the signal's duration is a power of two: $N = 2^l$. Consider what happens to the even-numbered and odd-numbered elements of the sequence in the DFT calculation.

$$\begin{aligned}
 S(k) &= s(0) + s(2) e^{(-j) \frac{2\pi 2k}{N}} + \cdots + s(N-2) e^{(-j) \frac{2\pi(N-2)k}{N}} + & (8.1) \\
 s(1) e^{(-j) \frac{2\pi k}{N}} + s(3) e^{(-j) \frac{2\pi(2+1)k}{N}} + \cdots + s(N-1) e^{(-j) \frac{2\pi(N-2+1)k}{N}} & = \\
 s(0) + s(2) e^{(-j) \frac{2\pi k}{\frac{N}{2}}} + \cdots + s(N-2) e^{(-j) \frac{2\pi(\frac{N}{2}-1)k}{\frac{N}{2}}} + & \\
 \left(s(1) + s(3) e^{(-j) \frac{2\pi k}{\frac{N}{2}}} + \cdots + s(N-1) e^{(-j) \frac{2\pi(\frac{N}{2}-1)k}{\frac{N}{2}}} \right) e^{-\frac{(j2\pi k)}{N}} &
 \end{aligned}$$

Each term in square brackets has the **form** of a $\frac{N}{2}$ -length DFT. The first one is a DFT of the even-numbered elements, and the second of the odd-numbered elements. The first DFT is combined with the second multiplied by the complex exponential $e^{-\frac{(j2\pi k)}{N}}$. The half-length transforms are each evaluated at frequency indices $k \in \{0, \dots, N-1\}$. Normally, the number of frequency indices in a DFT calculation range between zero and the transform length minus one. The **computational advantage** of the FFT comes from recognizing the periodic nature of the discrete Fourier transform. The FFT simply reuses the computations made in the half-length transforms and combines them through additions and the multiplication by $e^{-\frac{(j2\pi k)}{N}}$, which is not periodic over $\frac{N}{2}$, to rewrite the length- N DFT. Figure 8.2 (Length-8 DFT decomposition) illustrates this decomposition. As it stands, we now compute two length- $\frac{N}{2}$ transforms (complexity $2O\left(\frac{N^2}{4}\right)$), multiply one of them by the complex exponential (complexity $O(N)$), and add the results (complexity $O(N)$). At this point, the total complexity is still dominated by the half-length DFT calculations, but the proportionality coefficient has been reduced.

Now for the fun. Because $N = 2^l$, each of the half-length transforms can be reduced to two quarter-length transforms, each of these to two eighth-length ones, etc. This decomposition continues until we are left with length-2 transforms. This transform is quite simple, involving only additions. Thus, the first stage of the FFT has $\frac{N}{2}$ length-2 transforms (see the bottom part of Figure 8.2 (Length-8 DFT decomposition)). Pairs of these transforms are combined by adding one to the other multiplied by a complex exponential. Each pair requires 4 additions and 4 multiplications, giving a total number of computations equaling $8\frac{N}{4} = \frac{N}{2}$. This number of computations does not change from stage to stage. Because the number of stages, the number of times the length can be divided by two, equals $\log_2 N$, the complexity of the FFT is $O(N \log N)$.

⁷This content is available online at <<http://cnx.org/content/m0528/2.7/>>.

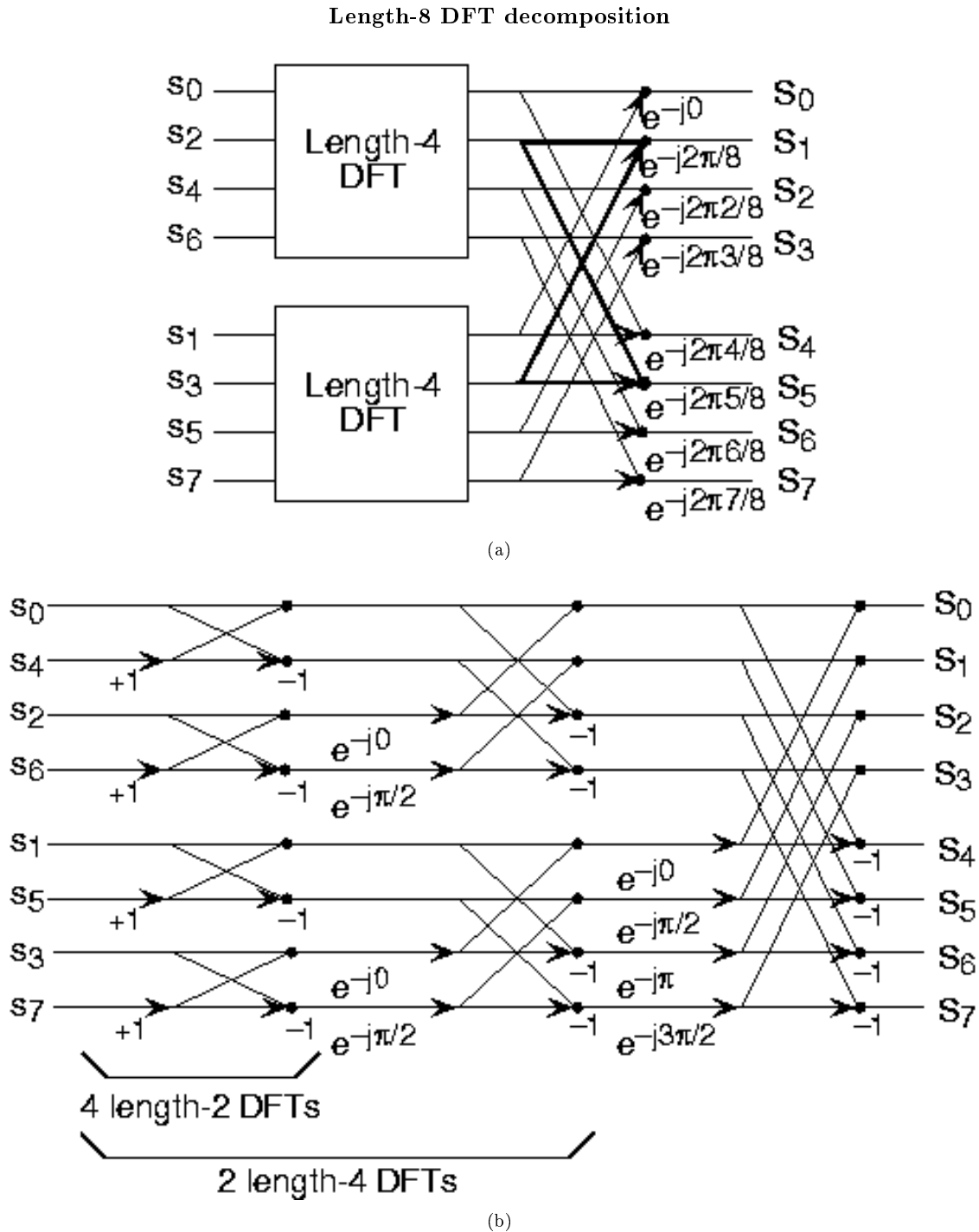


Figure 8.2: The initial decomposition of a length-8 DFT into the terms using even- and odd-indexed inputs marks the first phase of developing the FFT algorithm. When these half-length transforms are successively decomposed, we are left with the diagram shown in the bottom panel that depicts the length-8 FFT computation.

Doing an example will make computational savings more obvious. Let's look at the details of a length-8 DFT. As shown on Figure 8.2 (Length-8 DFT decomposition), we first decompose the DFT into two length-4 DFTs, with the outputs added and subtracted together in pairs. Considering Figure 8.2 (Length-8 DFT decomposition) as the frequency index goes from 0 through 7, we recycle values from the length-4 DFTs into the final calculation because of the periodicity of the DFT output. Examining how pairs of outputs are collected together, we create the basic computational element known as a **butterfly** (Figure 8.3 (Butterfly)).

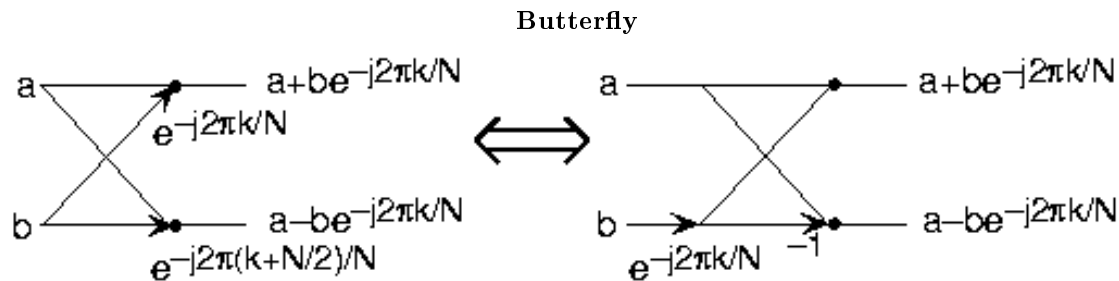


Figure 8.3: The basic computational element of the fast Fourier transform is the butterfly. It takes two complex numbers, represented by a and b , and forms the quantities shown. Each butterfly requires one complex multiplication and two complex additions.

By considering together the computations involving common output frequencies from the two half-length DFTs, we see that the two complex multiplies are related to each other, and we can reduce our computational work even further. By further decomposing the length-4 DFTs into two length-2 DFTs and combining their outputs, we arrive at the diagram summarizing the length-8 fast Fourier transform (Figure 8.2 (Length-8 DFT decomposition)). Although most of the complex multiplies are quite simple (multiplying by $e^{-j\pi}$ means negating real and imaginary parts), let's count those for purposes of evaluating the complexity as full complex multiplies. We have $\frac{N}{2} = 4$ complex multiplies and $2N = 16$ additions for each stage and $\log_2 N = 3$ stages, making the number of basic computations $\frac{3N}{2} \log_2 N$ as predicted.

Exercise 8.2

(Solution on p. 170.)

Note that the ordering of the input sequence in the two parts of Figure 8.2 (Length-8 DFT decomposition) aren't quite the same. Why not? How is the ordering determined?

Other "fast" algorithms were discovered, all of which make use of how many common factors the transform length N has. In number theory, the number of prime factors a given integer has measures how **composite** it is. The numbers 16 and 81 are highly composite (equaling 2^4 and 3^4 respectively), the number 18 is less so ($2^1 3^2$), and 17 not at all (it's prime). In over thirty years of Fourier transform algorithm development, the original Cooley-Tukey algorithm is far and away the most frequently used. It is so computationally efficient that power-of-two transform lengths are frequently used regardless of what the actual length of the data.

Solutions to Exercises in Chapter 8

Solution to Exercise 8.1 (p. 165)

When the signal is real-valued, we may only need half the spectral values, but the complexity remains unchanged. If the data are complex-valued, which demands retaining all frequency values, the complexity is again the same. When only K frequencies are needed, the complexity is $O(KN)$.

Solution to Exercise 8.2 (p. 169)

The upper panel has not used the FFT algorithm to compute the length-4 DFTs while the lower one has. The ordering is determined by the algorithm.

Chapter 9

Convergence

9.1 Convergence of Sequences¹

9.1.1 What is a Sequence?

Definition 9: sequence

A sequence is a function g_n defined on the positive integers 'n'. We often denote a sequence by $\{g_n\}_{n=1}^{\infty}$

Example

A real number sequence:

$$g_n = \frac{1}{n}$$

Example

A vector sequence:

$$g_n = \begin{pmatrix} \sin\left(\frac{n\pi}{2}\right) \\ \cos\left(\frac{n\pi}{2}\right) \end{pmatrix}$$

Example

A function sequence:

$$g_n(t) = \begin{cases} 1 & \text{if } 0 \leq t < \frac{1}{n} \\ 0 & \text{otherwise} \end{cases}$$

NOTE: A function can be thought of as an infinite dimensional vector where for each value of 't' we have one dimension

9.1.2 Convergence of Real Sequences

Definition 10: limit

A sequence $\{g_n\}_{n=1}^{\infty}$ converges to a limit $g \in \mathbb{R}$ if for every $\epsilon > 0$ there is an integer N such that

$$|g_i - g| < \epsilon, \quad i \geq N$$

¹This content is available online at <http://cnx.org/content/m10883/2.4/>.

We usually denote a limit by writing

$$\lim_{i \rightarrow \infty} g_i = g$$

or

$$g_i \rightarrow g$$

The above definition means that no matter how small we make ϵ , except for a finite number of g_i 's, all points of the sequence are within distance ϵ of g .

Example 9.1

We are given the following convergent sequence:

$$g_n = \frac{1}{n} \tag{9.1}$$

Intuitively we can assume the following limit:

$$\lim_{n \rightarrow \infty} g_n = 0$$

Let us prove this rigorously. Say that we are given a real number $\epsilon > 0$. Let us choose $N = \lceil \frac{1}{\epsilon} \rceil$, where $\lceil x \rceil$ denotes the smallest integer larger than x . Then for $n \geq N$ we have

$$|g_n - 0| = \frac{1}{n} \leq \frac{1}{N} < \epsilon$$

Thus,

$$\lim_{n \rightarrow \infty} g_n = 0$$

Example 9.2

Now let us look at the following non-convergent sequence

$$g_n = \begin{cases} 1 & \text{if } n = \text{even} \\ -1 & \text{if } n = \text{odd} \end{cases}$$

This sequence oscillates between 1 and -1, so it will therefore never converge.

9.1.2.1 Problems

For practice, say which of the following sequences converge and give their limits if they exist.

1. $g_n = n$
2. $g_n = \begin{cases} \frac{1}{n} & \text{if } n = \text{even} \\ \frac{-1}{n} & \text{if } n = \text{odd} \end{cases}$
3. $g_n = \begin{cases} \frac{1}{n} & \text{if } n \neq \text{power of } 10 \\ 1 & \text{otherwise} \end{cases}$
4. $g_n = \begin{cases} n & \text{if } n < 10^5 \\ \frac{1}{n} & \text{if } n \geq 10^5 \end{cases}$
5. $g_n = \sin\left(\frac{\pi}{n}\right)$
6. $g_n = j^n$

9.2 Convergence of Vectors²

9.2.1 Convergence of Vectors

We now discuss pointwise and norm convergence of vectors. Other types of convergence also exist, and one in particular, uniform convergence (Section 9.3), can also be studied. For this discussion, we will assume that the vectors belong to a normed vector space (Section 15.2).

9.2.1.1 Pointwise Convergence

A sequence (Section 9.1) $\{g_n\}_{n=1}^{\infty}$ converges **pointwise** to the limit \mathbf{g} if each element of g_n converges to the corresponding element in \mathbf{g} . Below are few examples to try and help illustrate this idea.

Example 9.3

$$g_n = \begin{pmatrix} g_n [1] \\ g_n [2] \end{pmatrix} = \begin{pmatrix} 1 + \frac{1}{n} \\ 2 - \frac{1}{n} \end{pmatrix}$$

First we find the following limits for our two g_n 's:

$$\lim_{n \rightarrow \infty} (g_n [1]) = 1$$

$$\lim_{n \rightarrow \infty} (g_n [2]) = 2$$

Therefore we have the following,

$$\lim_{n \rightarrow \infty} g_n = \mathbf{g}$$

pointwise, where $\mathbf{g} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Example 9.4

$$g_n(t) = \frac{t}{n}, \quad t \in \mathbb{R}$$

As done above, we first want to examine the limit

$$\lim_{n \rightarrow \infty} g_n(t_0) = \lim_{n \rightarrow \infty} \frac{t_0}{n} = 0$$

where $t_0 \in \mathbb{R}$. Thus $\lim_{n \rightarrow \infty} g_n = g$ pointwise where $g(t) = 0$ for all $t \in \mathbb{R}$.

9.2.1.2 Norm Convergence

The sequence (Section 9.1) $\{g_n\}_{n=1}^{\infty}$ converges to \mathbf{g} in norm if $\lim_{n \rightarrow \infty} \|g_n - \mathbf{g}\| = 0$. Here $\|\cdot\|$ is the norm (Section 15.2) of the corresponding vector space of g_n 's. Intuitively this means the distance between vectors g_n and \mathbf{g} decreases to 0.

Example 9.5

$$g_n = \begin{pmatrix} 1 + \frac{1}{n} \\ 2 - \frac{1}{n} \end{pmatrix}$$

²This content is available online at <<http://cnx.org/content/m10894/2.2/>>.

$$\text{Let } \mathbf{g} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\begin{aligned} \|g_n - \mathbf{g}\| &= \sqrt{\left(1 + \frac{1}{n} - 1\right)^2 + (-1)^2} \\ &= \sqrt{\frac{1}{n^2} + \frac{1}{n^2}} \\ &= \frac{\sqrt{2}}{n} \end{aligned} \tag{9.2}$$

Thus $\lim_{n \rightarrow \infty} \|g_n - \mathbf{g}\| = 0$ Therefore, $g_n \rightarrow \mathbf{g}$ in norm.

Example 9.6

$$g_n(t) = \begin{cases} \frac{t}{n} & \text{if } 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Let $g(t) = 0$ for all t .

$$\begin{aligned} \|g_n(t) - g(t)\| &= \int_0^1 \frac{t^2}{n^2} dt \\ &= \frac{t^3}{3n^2} \Big|_{n=0}^1 \\ &= \frac{1}{3n^2} \end{aligned} \tag{9.3}$$

Thus $\lim_{n \rightarrow \infty} \|g_n(t) - g(t)\| = 0$ Therefore, $g_n(t) \rightarrow g(t)$ in norm.

9.2.2 Pointwise vs. Norm Convergence

Theorem 9.1:

For \mathbb{R}^m , pointwise and norm convergence are equivalent.

Proof: Pointwise \Rightarrow Norm

$$g_n[i] \rightarrow g[i]$$

Assuming the above, then

$$\|g_n - \mathbf{g}\|^2 = \sum_{i=1}^m \left((g_n[i] - g[i])^2 \right)$$

Thus,

$$\begin{aligned} \lim_{n \rightarrow \infty} (\|g_n - \mathbf{g}\|)^2 &= \lim_{n \rightarrow \infty} \sum_{i=1}^m 2 \\ &= \sum_{i=1}^m \left(\lim_{n \rightarrow \infty} 2 \right) \\ &= 0 \end{aligned} \tag{9.4}$$

Proof: Norm \Rightarrow Pointwise

$$\|g_n - \mathbf{g}\| \rightarrow 0$$

$$\begin{aligned} \lim_{n \rightarrow \infty} \sum_{i=1}^m 2 &= \sum_{i=1}^m \left(\lim_{n \rightarrow \infty} 2 \right) \\ &= 0 \end{aligned} \tag{9.5}$$

Since each term is greater than or equal zero, all 'm' terms must be zero. Thus,

$$\lim_{n \rightarrow \infty} 2 = 0$$

for all i . Therefore,

$$g_n \rightarrow g \text{ pointwise}$$

NOTE: In infinite dimensional spaces the above theorem is no longer true. We prove this with counter examples shown below.

9.2.2.1 Counter Examples

Example 9.7: Pointwise [U+21CF] Norm

We are given the following function:

$$g_n(t) = \begin{cases} n & \text{if } 0 < t < \frac{1}{n} \\ 0 & \text{otherwise} \end{cases}$$

Then $\lim_{n \rightarrow \infty} g_n(t) = 0$ This means that,

$$g_n(t) \rightarrow g(t)$$

where for all t $g(t) = 0$.

Now,

$$\begin{aligned} (\|g_n\|)^2 &= \int_{-\infty}^{\infty} (|g_n(t)|)^2 dt \\ &= \int_0^{\frac{1}{n}} n^2 dt \\ &= n \rightarrow \infty \end{aligned} \tag{9.6}$$

Since the function norms blow up, they cannot converge to any function with finite norm.

Example 9.8: Norm [U+21CF] Pointwise

We are given the following function:

$$g_n(t) = \begin{cases} 1 & \text{if } 0 < t < \frac{1}{n} \text{ if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$

$$g_n(t) = \begin{cases} -1 & \text{if } 0 < t < \frac{1}{n} \text{ if } n \text{ is odd} \\ 0 & \text{otherwise} \end{cases}$$

Then,

$$\|g_n - g\| = \int_0^{\frac{1}{n}} 1 dt = \frac{1}{n} \rightarrow 0$$

where $g(t) = 0$ for all t . Therefore,

$$g_n \rightarrow g \text{ in norm}$$

However, at $t = 0$, $g_n(t)$ oscillates between -1 and 1, and so it does not converge. Thus, $g_n(t)$ does not converge pointwise.

9.2.2.2 Problems

Prove if the following sequences are pointwise convergent, norm convergent, or both and then state their limits.

$$1. g_n(t) = \begin{cases} \frac{1}{nt} & \text{if } 0 < t \\ 0 & \text{if } t \leq 0 \end{cases}$$

$$2. g_n(t) = \begin{cases} e^{-(nt)} & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases}$$

9.3 Uniform Convergence of Function Sequences³

9.3.1 Uniform Convergence of Function Sequences

For this discussion, we will only consider functions with g_n where

$$\mathbb{R} \rightarrow \mathbb{R}$$

Definition 11: Uniform Convergence

The sequence (Section 9.1) $\{g_n\}_{n=1}^{\infty}$ converges uniformly to function g if for every $\epsilon > 0$ there is an integer N such that $n \geq N$ implies

$$|g_n(t) - g(t)| \leq \epsilon \tag{9.7}$$

for all $t \in \mathbb{R}$.

Obviously every uniformly convergent sequence is pointwise (Section 9.2) convergent. The difference between pointwise and uniform convergence is this: If $\{g_n\}$ converges pointwise to g , then for every $\epsilon > 0$ and for every $t \in \mathbb{R}$ there is an integer N depending on ϵ and t such that (9.7) holds if $n \geq N$. If $\{g_n\}$ converges uniformly to g , it is possible for each $\epsilon > 0$ to find one integer N that will do for all $t \in \mathbb{R}$.

Example 9.9

$$g_n(t) = \frac{1}{n}, \quad t \in \mathbb{R}$$

Let $\epsilon > 0$ be given. Then choose $N = \lceil \frac{1}{\epsilon} \rceil$. Obviously,

$$|g_n(t) - 0| \leq \epsilon, \quad n \geq N$$

for all t . Thus, $g_n(t)$ converges uniformly to 0.

Example 9.10

$$g_n(t) = \frac{t}{n}, \quad t \in \mathbb{R}$$

Obviously for any $\epsilon > 0$ we cannot find a single function $g_n(t)$ for which (9.7) holds with $g(t) = 0$ for all t . Thus g_n is not uniformly convergent. However we do have:

$$g_n(t) \rightarrow g(t) \text{ pointwise}$$

CONCLUSION: Uniform convergence always implies pointwise convergence, but pointwise convergence does not guarantee uniform convergence.

³This content is available online at <http://cnx.org/content/m10895/2.5/>.

9.3.1.1 Problems

Rigorously prove if the following functions converge pointwise, uniformly, or both.

1. $g_n(t) = \frac{\sin(t)}{n}$

2. $g_n(t) = e^{\frac{t}{n}}$

3. $g_n(t) = \begin{cases} \frac{1}{nt} & \text{if } t > 0 \\ 0 & \text{if } t \leq 0 \end{cases}$

Chapter 10

Discrete Time Fourier Transform (DTFT)

10.1 Discrete Fourier Transformation¹

10.1.1 N-point Discrete Fourier Transform (DFT)

$$X[k] = \sum_{n=0}^{N-1} \left(x[n] e^{(-j) \frac{2\pi}{N} kn} \right) , \quad k = \{0, \dots, N-1\} \quad (10.1)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} \left(X[k] e^{j \frac{2\pi}{N} kn} \right) , \quad n = \{0, \dots, N-1\} \quad (10.2)$$

Note that:

- $X[k]$ is the DTFT evaluated at $\omega = \frac{2\pi}{N}k$, $k = \{0, \dots, N-1\}$
- Zero-padding $x[n]$ to M samples prior to the DFT yields an M -point uniform sampled version of the DTFT:

$$X \left(e^{j \frac{2\pi}{M} k} \right) = \sum_{n=0}^{N-1} \left(x[n] e^{(-j) \frac{2\pi}{M} k n} \right) \quad (10.3)$$

$$X \left(e^{j \frac{2\pi}{M} k} \right) = \sum_{n=0}^{N-1} \left(x_{zp}[n] e^{(-j) \frac{2\pi}{M} k n} \right)$$

$$X \left(e^{j \frac{2\pi}{M} k} \right) = X_{zp}[k] , \quad k = \{0, \dots, M-1\}$$

- The N -pt DFT is sufficient to reconstruct the entire DTFT of an N -pt sequence:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} \left(x[n] e^{(-j)\omega n} \right) \quad (10.4)$$

$$X(e^{j\omega}) = \left(\sum_{n=0}^{N-1} \left(\frac{1}{N} \right) \right) \sum_{k=0}^{N-1} \left(X[k] e^{j \frac{2\pi}{N} kn} e^{(-j)\omega n} \right)$$

¹This content is available online at <<http://cnx.org/content/m10421/2.10/>>.

$$X(e^{j\omega}) = \left(\sum_{k=0}^{N-1} X[k] \right) \frac{1}{N} \sum_{k=0}^{N-1} \left(e^{(-j)(\omega - \frac{2\pi}{N}k)n} \right)$$

$$X(e^{j\omega}) = \left(\sum_{k=0}^{N-1} X[k] \right) \frac{1}{N} \left(\frac{\sin\left(\frac{\omega N - 2\pi k}{2}\right)}{\sin\left(\frac{\omega N - 2\pi k}{2N}\right)} e^{(-j)(\omega - \frac{2\pi}{N}k)\frac{N-1}{2}} \right)$$

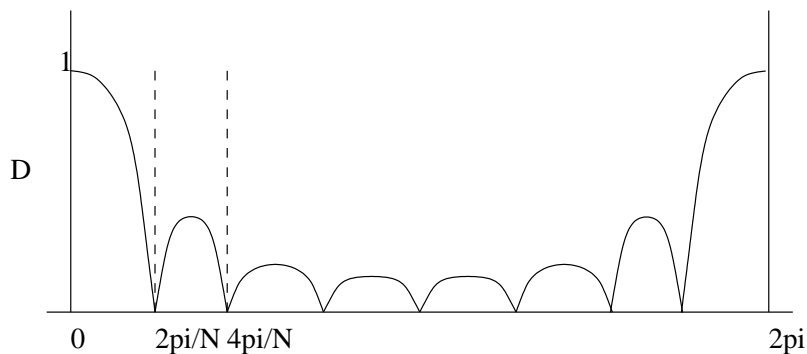


Figure 10.1: Dirichlet sinc, $\frac{1}{N} \frac{\sin(\frac{\omega N}{2})}{\sin(\frac{\omega}{2})}$

- The DFT has a convenient matrix representation. Defining $W_N = e^{(-j)\frac{2\pi}{N}}$,

$$\begin{pmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{pmatrix} = \begin{pmatrix} W_N^0 & W_N^0 & W_N^0 & W_N^0 & \dots \\ W_N^0 & W_N^1 & W_N^2 & W_N^3 & \dots \\ W_N^0 & W_N^2 & W_N^4 & W_N^6 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{pmatrix} \quad (10.5)$$

where $X = W(x)$ respectively. W has the following properties:

- W is Vandermonde: the n th column of W is a polynomial in W_N^n
 - W is symmetric: $W = W^T$
 - $\frac{1}{\sqrt{N}}W$ is unitary: $\left(\frac{1}{\sqrt{N}}W\right)\left(\frac{1}{\sqrt{N}}W\right)^H = \left(\frac{1}{\sqrt{N}}W\right)^H\left(\frac{1}{\sqrt{N}}W\right) = I$
 - $\frac{1}{N}W^* = W^{-1}$, the IDFT matrix.
- For N a power of 2, the FFT can be used to compute the DFT using about $\frac{N}{2}\log_2 N$ rather than N^2 operations.

N	$\frac{N}{2}\log_2 N$	N^2
16	32	256
64	192	4096
256	1024	65536
1024	5120	1048576

10.2 Discrete Fourier Transform (DFT)²

The discrete-time Fourier transform (and the continuous-time transform as well) can be evaluated when we have an analytic expression for the signal. Suppose we just have a signal, such as the speech signal used in the previous chapter, for which there is no formula. How then would you compute the spectrum? For example, how did we compute a spectrogram such as the one shown in the speech signal example³? The Discrete Fourier Transform (DFT) allows the computation of spectra from discrete-time data. While in discrete-time we can *exactly* calculate spectra, for analog signals no similar exact spectrum computation exists. For analog-signal spectra, we must build special devices, which turn out in most cases to consist of A/D converters and discrete-time computations. Certainly discrete-time spectral analysis is more flexible than continuous-time spectral analysis.

The formula for the DTFT⁴ is a sum, which conceptually can be easily computed save for two issues.

- *Signal duration.* The sum extends over the signal's duration, which must be finite to compute the signal's spectrum. It is exceedingly difficult to store an infinite-length signal in any case, so we'll assume that the signal extends over $[0, N - 1]$.
- *Continuous frequency.* Subtler than the signal duration issue is the fact that the frequency variable is continuous: It may only need to span one period, like $[-\frac{1}{2}, \frac{1}{2}]$ or $[0, 1]$, but the DTFT formula as it stands requires evaluating the spectra at *all* frequencies within a period. Let's compute the spectrum at a few frequencies; the most obvious ones are the equally spaced ones $f = \frac{k}{K}$, $k \in \{0, \dots, K - 1\}$.

We thus define the **discrete Fourier transform** (DFT) to be

$$S(k) = \sum_{n=0}^{N-1} \left(s(n) e^{-j\frac{2\pi nk}{K}} \right) \quad , \quad k \in \{0, \dots, K - 1\} \quad (10.6)$$

Here, $S(k)$ is shorthand for $S\left(e^{j2\pi\frac{k}{K}}\right)$.

We can compute the spectrum at as many equally spaced frequencies as we like. Note that you can think about this computationally motivated choice as **sampling** the spectrum; more about this interpretation later. The issue now is how many frequencies are enough to capture how the spectrum changes with frequency. One way of answering this question is determining an inverse discrete Fourier transform formula: given $S(k)$, $k = \{0, \dots, K - 1\}$ how do we find $s(n)$, $n = \{0, \dots, N - 1\}$? Presumably, the formula will be of the form $s(n) = \sum_{k=0}^{K-1} \left(S(k) e^{j\frac{2\pi nk}{K}} \right)$. Substituting the DFT formula in this prototype inverse transform yields

$$s(n) = \sum_{k=0}^{K-1} \left(\sum_{m=0}^{N-1} \left(s(m) e^{-j\frac{2\pi mk}{K}} e^{j\frac{2\pi nk}{K}} \right) \right) \quad (10.7)$$

Note that the orthogonality relation we use so often has a different character now.

$$\sum_{k=0}^{K-1} \left(e^{-j\frac{2\pi km}{K}} e^{j\frac{2\pi kn}{K}} \right) = \begin{cases} K & \text{if } m = \{n, (n \pm K), (n \pm 2K), \dots\} \\ 0 & \text{otherwise} \end{cases} \quad (10.8)$$

We obtain nonzero value whenever the two indices differ by multiples of K . We can express this result as $K \sum_l (\delta(m - n - lK))$. Thus, our formula becomes

$$s(n) = \sum_{m=0}^{N-1} \left(s(m) K \sum_{l=-\infty}^{\infty} (\delta(m - n - lK)) \right) \quad (10.9)$$

²This content is available online at <http://cnx.org/content/m10249/2.26/>.

³"Modeling the Speech Signal", Figure 5: spectrogram <http://cnx.org/content/m0049/latest/#spectrogram>

⁴"Discrete-Time Fourier Transform (DTFT)", (1) <http://cnx.org/content/m10247/latest/#eqn1>

The integers n and m both range over $\{0, \dots, N-1\}$. To have an inverse transform, we need the sum to be a *single* unit sample for m, n in this range. If it did not, then $s(n)$ would equal a sum of values, and we would not have a valid transform: Once going into the frequency domain, we could not get back unambiguously! Clearly, the term $l = 0$ always provides a unit sample (we'll take care of the factor of K soon). If we evaluate the spectrum at *fewer* frequencies than the signal's duration, the term corresponding to $m = n + K$ will also appear for some values of $m, n = \{0, \dots, N-1\}$. This situation means that our prototype transform equals $s(n) + s(n + K)$ for some values of n . The only way to eliminate this problem is to require $K \geq N$: We *must* have at least as many frequency samples as the signal's duration. In this way, we can return from the frequency domain we entered via the DFT.

Exercise 10.1**(Solution on p. 190.)**

When we have fewer frequency samples than the signal's duration, some discrete-time signal values equal the sum of the original signal values. Given the sampling interpretation of the spectrum, characterize this effect a different way.

Another way to understand this requirement is to use the theory of linear equations. If we write out the expression for the DFT as a set of linear equations,

$$s(0) + s(1) + \dots + s(N-1) = S(0) \quad (10.10)$$

$$s(0) + s(1)e^{(-j)\frac{2\pi}{K}} + \dots + s(N-1)e^{(-j)\frac{2\pi(N-1)}{K}} = S(1)$$

$$\vdots$$

$$s(0) + s(1)e^{(-j)\frac{2\pi(K-1)}{K}} + \dots + s(N-1)e^{(-j)\frac{2\pi(N-1)(K-1)}{K}} = S(K-1)$$

we have K equations in N unknowns if we want to find the signal from its sampled spectrum. This requirement is impossible to fulfill if $K < N$; we must have $K \geq N$. Our orthogonality relation essentially says that if we have a sufficient number of equations (frequency samples), the resulting set of equations can indeed be solved.

By convention, the number of DFT frequency values K is chosen to equal the signal's duration N . The discrete Fourier transform pair consists of

Discrete Fourier Transform Pair

$$S(k) = \sum_{n=0}^{N-1} \left(s(n) e^{-j\frac{2\pi nk}{N}} \right) \quad (10.11)$$

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left(S(k) e^{j\frac{2\pi nk}{N}} \right) \quad (10.12)$$

Example 10.1

Use this demonstration to perform DFT analysis of a signal.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10249/latest/DFTanalysis.llb>

Example 10.2

Use this demonstration to synthesize a signal from a DFT sequence.

This is an unsupported media type. To view, please see
http://cnx.org/content/m10249/latest/DFT_Component_Manipulation.llb

10.3 Table of Common Fourier Transforms⁵

Time Domain Signal	Frequency Domain Signal	Condition
$e^{-(at)}u(t)$	$\frac{1}{a+j\omega}$	$a > 0$
$e^{at}u(-t)$	$\frac{1}{a-j\omega}$	$a > 0$
$e^{-a t }$	$\frac{2a}{a^2+\omega^2}$	$a > 0$
$te^{-(at)}u(t)$	$\frac{1}{(a+j\omega)^2}$	$a > 0$
$t^n e^{-(at)}u(t)$	$\frac{n!}{(a+j\omega)^{n+1}}$	$a > 0$
$\delta(t)$	1	
1	$2\pi\delta(\omega)$	
$e^{j\omega_0 t}$	$2\pi\delta(\omega - \omega_0)$	
$\cos(\omega_0 t)$	$\pi(\delta(\omega - \omega_0) + \delta(\omega + \omega_0))$	
$\sin(\omega_0 t)$	$j\pi(\delta(\omega + \omega_0) - \delta(\omega - \omega_0))$	
$u(t)$	$\pi\delta(\omega) + \frac{1}{j\omega}$	
$\text{sgn}(t)$	$\frac{2}{j\omega}$	
$\cos(\omega_0 t)u(t)$	$\frac{\pi}{2}(\delta(\omega - \omega_0) + \delta(\omega + \omega_0)) + \frac{j\omega}{\omega_0^2 - \omega^2}$	
$\sin(\omega_0 t)u(t)$	$\frac{\pi}{2j}(\delta(\omega - \omega_0) - \delta(\omega + \omega_0)) + \frac{\omega_0}{\omega_0^2 - \omega^2}$	
$e^{-(at)}\sin(\omega_0 t)u(t)$	$\frac{\omega_0}{(a+j\omega)^2 + \omega_0^2}$	$a > 0$
$e^{-(at)}\cos(\omega_0 t)u(t)$	$\frac{a+j\omega}{(a+j\omega)^2 + \omega_0^2}$	$a > 0$
$u(t + \tau) - u(t - \tau)$	$2\tau \frac{\sin(\omega\tau)}{\omega\tau} = 2\tau \text{sinc}(\omega\tau)$	
$\frac{\omega_0}{\pi} \frac{\sin(\omega_0 t)}{\omega_0 t} = \frac{\omega_0}{\pi} \text{sinc}(\omega_0 t)$	$u(\omega + \omega_0) - u(\omega - \omega_0)$	
$\left(\frac{t}{\tau} + 1\right)\left(u\left(\frac{t}{\tau} + 1\right) - u\left(\frac{t}{\tau}\right)\right) + \left(-\frac{t}{\tau} + 1\right)\left(u\left(\frac{t}{\tau}\right) - u\left(\frac{t}{\tau} - 1\right)\right) = \text{triang}\left(\frac{t}{2\tau}\right)$	$\tau \text{sinc}^2\left(\frac{\omega\tau}{2}\right)$	
$\frac{\omega_0}{2\pi} \text{sinc}^2\left(\frac{\omega_0 t}{2}\right)$	$\left(\frac{\omega}{\omega_0} + 1\right)\left(u\left(\frac{\omega}{\omega_0} + 1\right) - u\left(\frac{\omega}{\omega_0}\right)\right) + \left(-\frac{\omega}{\omega_0} + 1\right)\left(u\left(\frac{\omega}{\omega_0}\right) - u\left(\frac{\omega}{\omega_0} - 1\right)\right) = \text{triang}\left(\frac{\omega}{2\omega_0}\right)$	
$\sum_{n=-\infty}^{\infty} (\delta(t - nT))$	$\omega_0 \sum_{n=-\infty}^{\infty} (\delta(\omega - n\omega_0))$	$\omega_0 = \frac{2\pi}{T}$
$e^{-\left(\frac{t^2}{2\sigma^2}\right)}$	$\sigma\sqrt{2\pi}e^{-\left(\frac{\sigma^2\omega^2}{2}\right)}$	

⁵This content is available online at <<http://cnx.org/content/m10099/2.9/>>.

10.4 Discrete-Time Fourier Transform (DTFT)⁶

Discrete-Time Fourier Transform

$$X(\omega) = \sum_{n=-\infty}^{\infty} (x(n) e^{-j\omega n}) \quad (10.13)$$

Inverse Discrete-Time Fourier Transform

$$x(n) = \frac{1}{2\pi} \int_0^{2\pi} X(\omega) e^{j\omega n} d\omega \quad (10.14)$$

10.4.1 Relevant Spaces

The Discrete-Time Fourier Transform⁷ maps infinite-length, discrete-time signals in l^2 to finite-length (or 2π -periodic), continuous-frequency signals in L^2 .

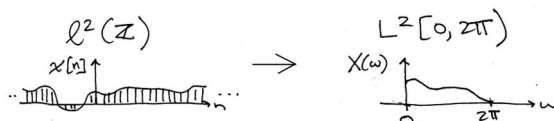


Figure 10.2: Mapping $l^2(\mathbb{Z})$ in the time domain to $L^2([0, 2\pi])$ in the frequency domain.

⁶This content is available online at <http://cnx.org/content/m10108/2.11/>.

⁷"Discrete-Time Fourier Transform (DTFT)" <http://cnx.org/content/m10247/latest/>

10.5 Discrete-Time Fourier Transform Properties⁸

Discrete-Time Fourier Transform Properties

	Sequence Domain	Frequency Domain
Linearity	$a_1 s_1(n) + a_2 s_2(n)$	$a_1 S_1(e^{j2\pi f}) + a_2 S_2(e^{j2\pi f})$
Conjugate Symmetry	$s(n)$ real	$S(e^{j2\pi f}) = S(e^{-j2\pi f})^*$
Even Symmetry	$s(n) = s(-n)$	$S(e^{j2\pi f}) = S(e^{-j2\pi f})$
Odd Symmetry	$s(n) = -s(-n)$	$S(e^{j2\pi f}) = -S(e^{-j2\pi f})$
Time Delay	$s(n - n_0)$	$e^{-j2\pi f n_0} S(e^{j2\pi f})$
Complex Modulation	$e^{j2\pi f_0 n} s(n)$	$S(e^{j2\pi(f-f_0)})$
Amplitude Modulation	$s(n) \cos(2\pi f_0 n)$	$\frac{S(e^{j2\pi(f-f_0)}) + S(e^{j2\pi(f+f_0)})}{2}$
	$s(n) \sin(2\pi f_0 n)$	$\frac{S(e^{j2\pi(f-f_0)}) - S(e^{j2\pi(f+f_0)})}{2j}$
Multiplication by n	$ns(n)$	$\frac{1}{-j2\pi} \frac{d}{df} (S(e^{j2\pi f}))$
Sum	$\sum_{n=-\infty}^{\infty} s(n)$	$S(e^{j2\pi 0})$
Value at Origin	$s(0)$	$\int_{-\frac{1}{2}}^{\frac{1}{2}} S(e^{j2\pi f}) df$
Parseval's Theorem	$\sum_{n=-\infty}^{\infty} (s(n) ^2)$	$\int_{-\frac{1}{2}}^{\frac{1}{2}} (S(e^{j2\pi f}) ^2) df$

Figure 10.3: Discrete-time Fourier transform properties and relations.

10.6 Discrete-Time Fourier Transform Pair⁹

When we obtain the discrete-time signal via sampling an analog signal, the Nyquist frequency corresponds to the discrete-time frequency $\frac{1}{2}$. To show this, note that a sinusoid at the Nyquist frequency $\frac{1}{2T_s}$ has a sampled waveform that equals

Sinusoid at Nyquist Frequency $1/2T$

$$\begin{aligned} \cos\left(2\pi \frac{1}{2T_s} n T_s\right) &= \cos(\pi n) \\ &= (-1)^n \end{aligned} \quad (10.15)$$

The exponential in the DTFT at frequency $\frac{1}{2}$ equals $e^{-\frac{j2\pi n}{2}} = e^{-j\pi n} = (-1)^n$, meaning that the correspondence between analog and discrete-time frequency is established:

Analog, Discrete-Time Frequency Relationship

$$f_D = f_A T_s \quad (10.16)$$

⁸This content is available online at <http://cnx.org/content/m0506/2.6/>.

⁹This content is available online at <http://cnx.org/content/m0525/2.6/>.

where f_D and f_A represent discrete-time and analog frequency variables, respectively. The aliasing figure¹⁰ provides another way of deriving this result. As the duration of each pulse in the periodic sampling signal $p_{T_s}(t)$ narrows, the amplitudes of the signal's spectral repetitions, which are governed by the Fourier series coefficients of $p_{T_s}(t)$, become increasingly equal.¹¹ Thus, the sampled signal's spectrum becomes periodic with period $\frac{1}{T_s}$. Thus, the Nyquist frequency $\frac{1}{2T_s}$ corresponds to the frequency $\frac{1}{2}$.

The inverse discrete-time Fourier transform is easily derived from the following relationship:

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-(j2\pi fm)} e^{+j\pi fn} df = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases} \quad (10.17)$$

Therefore, we find that

$$\begin{aligned} \int_{-\frac{1}{2}}^{\frac{1}{2}} S(e^{j2\pi f}) e^{+j2\pi fn} df &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_m (s(m) e^{-(j2\pi fm)} e^{+j2\pi fn}) df \\ &= \sum_m \left(s(m) \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-(j2\pi f)(m-n)} df \right) \\ &= s(n) \end{aligned} \quad (10.18)$$

The Fourier transform pairs in discrete-time are

Fourier Transform Pairs in Discrete Time

$$S(e^{j2\pi f}) = \sum_n (s(n) e^{-(j2\pi fn)}) \quad (10.19)$$

Fourier Transform Pairs in Discrete Time

$$s(n) = \int_{-\frac{1}{2}}^{\frac{1}{2}} S(e^{j2\pi f}) e^{+j2\pi fn} df \quad (10.20)$$

10.7 DTFT Examples¹²

Example 10.3

Let's compute the discrete-time Fourier transform of the exponentially decaying sequence $s(n) = a^n u(n)$, where $u(n)$ is the unit-step sequence. Simply plugging the signal's expression into the Fourier transform formula,

Fourier Transform Formula

$$\begin{aligned} S(e^{j2\pi f}) &= \sum_{n=-\infty}^{\infty} (a^n u(n) e^{-(j2\pi fn)}) \\ &= \sum_{n=0}^{\infty} (a^n e^{-(j2\pi fn)}) \end{aligned} \quad (10.21)$$

This sum is a special case of the **geometric series**.

Geometric Series

$$\sum_{n=0}^{\infty} (\alpha^n) = \frac{1}{1-\alpha}, \quad |\alpha| < 1 \quad (10.22)$$

¹⁰"The Sampling Theorem", Figure 2: aliasing <<http://cnx.org/content/m0050/latest/#alias>>

¹¹Examination of the periodic pulse signal reveals that as Δ decreases, the value of c_0 , the largest Fourier coefficient, decreases to zero: $|c_0| = \frac{A\Delta}{T}$. Thus, to maintain a mathematically viable Sampling Theorem, the amplitude A must increase as $\frac{1}{\Delta}$, becoming infinitely large as the pulse duration decreases. Practical systems use a small value of Δ , say $0.1T_s$ and use amplifiers to rescale the signal.

¹²This content is available online at <<http://cnx.org/content/m0524/2.10/>>.

Thus, as long as $|a| < 1$, we have our Fourier transform.

$$S(e^{j2\pi f}) = \frac{1}{1 - ae^{-j2\pi f}} \quad (10.23)$$

Using Euler's relation, we can express the magnitude and phase of this spectrum.

$$|S(e^{j2\pi f})| = \frac{1}{\sqrt{(1 - a\cos(2\pi f))^2 + a^2\sin^2(2\pi f)}} \quad (10.24)$$

$$\angle(S(e^{j2\pi f})) = -\left(\arctan\left(\frac{a\sin(2\pi f)}{1 - a\cos(2\pi f)}\right)\right) \quad (10.25)$$

No matter what value of a we choose, the above formulae clearly demonstrate the periodic nature of the spectra of discrete-time signals. Figure 10.4 shows indeed that the spectrum is a periodic function. We need only consider the spectrum between $-\frac{1}{2}$ and $\frac{1}{2}$ to unambiguously define it. When $a > 0$, we have a lowpass spectrum — the spectrum diminishes as frequency increases from 0 to $\frac{1}{2}$ — with increasing a leading to a greater low frequency content; for $a < 0$, we have a highpass spectrum (Figure 10.5).

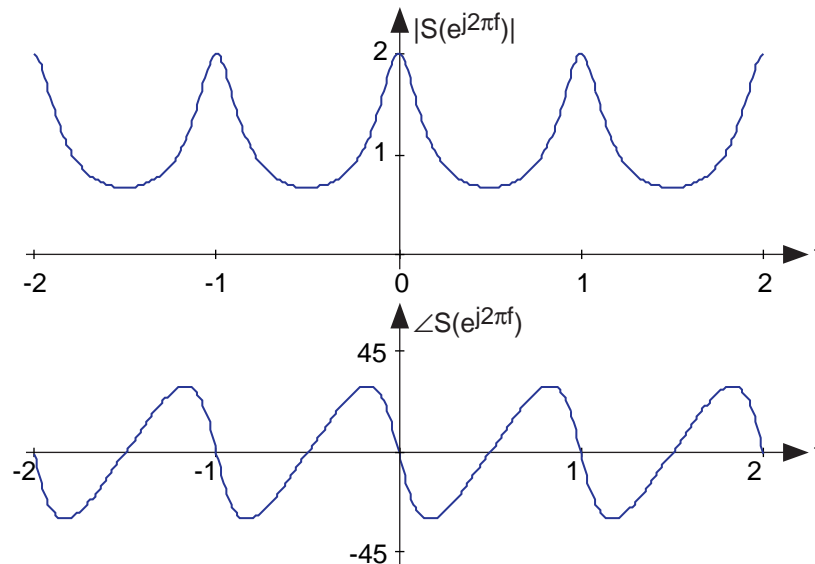


Figure 10.4: The spectrum of the exponential signal ($a = 0.5$) is shown over the frequency range $[-2, 2]$, clearly demonstrating the periodicity of all discrete-time spectra. The angle has units of degrees.

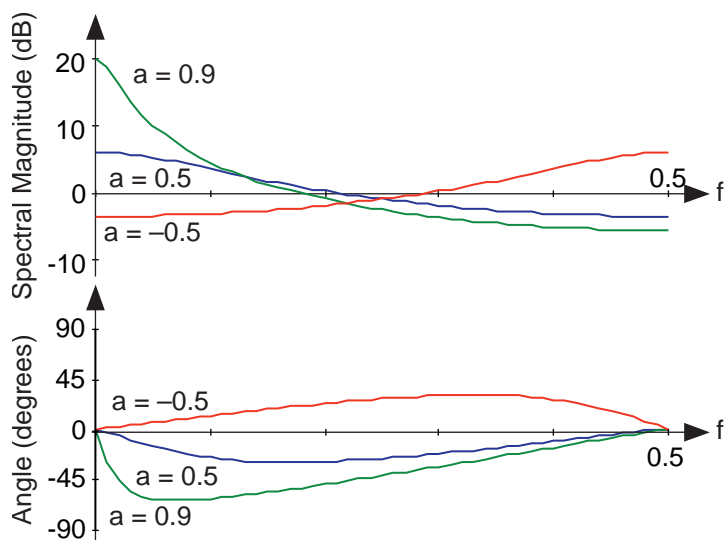


Figure 10.5: The spectra of several exponential signals are shown. What is the apparent relationship between the spectra for $a = 0.5$ and $a = -0.5$?

Example 10.4

Analogous to the analog pulse signal, let's find the spectrum of the length- N pulse sequence.

$$s(n) = \begin{cases} 1 & \text{if } 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \quad (10.26)$$

The Fourier transform of this sequence has the form of a truncated geometric series.

$$S(e^{j2\pi f}) = \sum_{n=0}^{N-1} \left(e^{-j2\pi f n} \right) \quad (10.27)$$

For the so-called finite geometric series, we know that

Finite Geometric Series

$$\sum_{n=n_0}^{N+n_0-1} (\alpha^n) = \alpha^{n_0} \frac{1 - \alpha^N}{1 - \alpha} \quad (10.28)$$

for all values of α .

Exercise 10.2

(Solution on p. 190.)

Derive this formula for the finite geometric series sum. The "trick" is to consider the difference between the series' sum and the sum of the series multiplied by α .

Applying this result yields (Figure 10.6.)

$$\begin{aligned} S(e^{j2\pi f}) &= \frac{1 - e^{-j2\pi f N}}{1 - e^{-j2\pi f}} \\ &= e^{-(j\pi f)(N-1)} \frac{\sin(\pi f N)}{\sin(\pi f)} \end{aligned} \quad (10.29)$$

The ratio of sine functions has the generic form of $\frac{\sin(Nx)}{\sin(x)}$, which is known as the **discrete-time sinc function**, $dsinc(x)$. Thus, our transform can be concisely expressed as $S(e^{j2\pi f}) = e^{-(j\pi f)(N-1)} dsinc(\pi f)$. The discrete-time pulse's spectrum contains many ripples, the number of which increase with N , the pulse's duration.

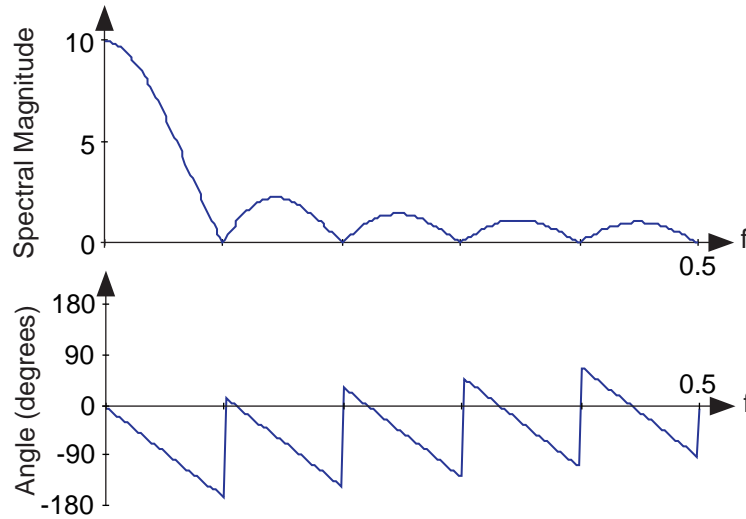


Figure 10.6: The spectrum of a length-ten pulse is shown. Can you explain the rather complicated appearance of the phase?

Solutions to Exercises in Chapter 10

Solution to Exercise 10.1 (p. 182)

This situation amounts to aliasing in the time-domain.

Solution to Exercise 10.2 (p. 188)

$$\alpha \sum_{n=n_0}^{N+n_0-1} (\alpha^n) - \sum_{n=n_0}^{N+n_0-1} (\alpha^n) = \alpha^{N+n_0} - \alpha^{n_0} \quad (10.30)$$

which, after manipulation, yields the geometric sum formula.

Chapter 11

Continuous Time Fourier Transform (CTFT)

11.1 Continuous-Time Fourier Transform (CTFT)¹

11.1.1 Introduction

Due to the large number of continuous-time signals that are present, the Fourier series² provided us the first glimpse of how we may represent some of these signals in a general manner: as a superposition of a number of sinusoids. Now, we can look at a way to represent continuous-time nonperiodic signals using the same idea of superposition. Below we will present the **Continuous-Time Fourier Transform (CTFT)**, also referred to as just the Fourier Transform (FT). Because the CTFT now deals with nonperiodic signals, we must now find a way to include *all* frequencies in the general equations.

11.1.1.1 Equations

Continuous-Time Fourier Transform

$$\mathcal{F}(\Omega) = \int_{-\infty}^{\infty} f(t) e^{-j\Omega t} dt \quad (11.1)$$

Inverse CTFT

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}(\Omega) e^{j\Omega t} d\Omega \quad (11.2)$$

WARNING: Do not be confused by notation - it is not uncommon to see the above formula written slightly different. One of the most common differences among many professors is the way that the exponential is written. Above we used the radial frequency variable Ω in the exponential, where $\Omega = 2\pi f$, but one will often see professors include the more explicit expression, $j2\pi ft$, in the exponential. Click here³ for an overview of the notation used in Connexion's DSP modules.

The above equations for the CTFT and its inverse come directly from the Fourier series and our understanding of its coefficients. For the CTFT we simply utilize integration rather than summation to be able to express the aperiodic signals. This should make sense since for the CTFT we are simply extending the

¹This content is available online at <<http://cnx.org/content/m10098/2.9/>>.

²"Classic Fourier Series" <<http://cnx.org/content/m0039/latest/>>

³"DSP Notation" <<http://cnx.org/content/m10161/latest/>>

ideas of the Fourier series to include nonperiodic signals, and thus the entire frequency spectrum. Look at the Derivation of the Fourier Transform⁴ for a more in depth look at this.

11.1.2 Relevant Spaces

The Continuous-Time Fourier Transform maps infinite-length, continuous-time signals in L^2 to infinite-length, continuous-frequency signals in L^2 . Review the Fourier Analysis (Section 7.1) for an overview of all the spaces used in Fourier analysis.

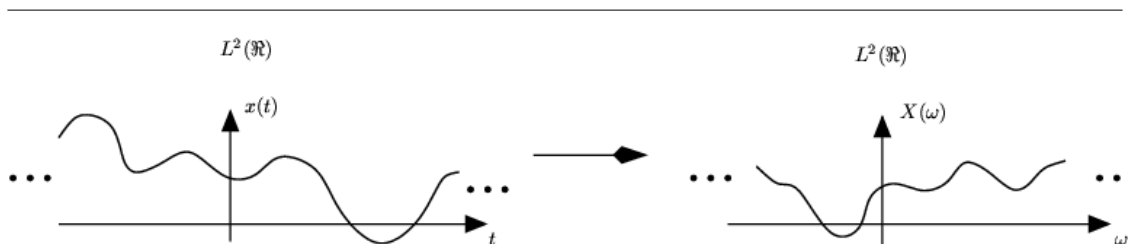


Figure 11.1: Mapping $L^2(\mathbb{R})$ in the time domain to $L^2(\mathbb{R})$ in the frequency domain.

For more information on the characteristics of the CTFT, please look at the module on Properties of the Fourier Transform (Section 11.2).

11.1.3 Example Problems

Exercise 11.1

Find the Fourier Transform (CTFT) of the function

(Solution on p. 196.)

$$f(t) = \begin{cases} e^{-\alpha t} & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (11.3)$$

Exercise 11.2

Find the inverse Fourier transform of the square wave defined as

(Solution on p. 196.)

$$X(\Omega) = \begin{cases} 1 & \text{if } |\Omega| \leq M \\ 0 & \text{otherwise} \end{cases} \quad (11.4)$$

11.2 Properties of the Continuous-Time Fourier Transform⁵

This module will look at some of the basic properties of the Continuous-Time Fourier Transform (Section 11.1) (CTFT). The first section contains a table that illustrates the properties, and the sections following it discuss a few of the more interesting properties in more depth. In the table, click on the operation name to be taken to the properties explanation found later on this page. Look at this module (Section 5.6) for an expanded table of more Fourier transform properties.

⁴"Derivation of the Fourier Transform" <<http://cnx.org/content/m0046/latest/>>

⁵This content is available online at <<http://cnx.org/content/m10100/2.13/>>.

NOTE: We will be discussing these properties for aperiodic, continuous-time signals but understand that very similar properties hold for discrete-time signals and periodic signals as well.

11.2.1 Table of CTFT Properties

Operation Name	Signal ($f(t)$)	Transform ($F(\omega)$)
Addition (Section 11.2.2.1: Linearity)	$f_1(t) + f_2(t)$	$F_1(\omega) + F_2(\omega)$
Scalar Multiplication (Section 11.2.2.1: Linearity)	$\alpha f(t)$	$\alpha F(\omega)$
Symmetry (Section 11.2.2.2: Symmetry)	$F(\omega)$	$2\pi f(-t)$
Time Scaling (Section 11.2.2.3: Time Scaling)	$f(\alpha t)$	$\frac{1}{ \alpha } F\left(\frac{\omega}{\alpha}\right)$
Time Shift (Section 11.2.2.4: Time Shifting)	$f(t - \tau)$	$F(\omega) e^{-j\omega\tau}$
Modulation (Frequency Shift) (Section 11.2.2.5: Modulation (Frequency Shift))	$f(t) e^{j\phi t}$	$F(\omega - \phi)$
Convolution in Time (Section 11.2.2.6: Convolution)	$(f_1(t), f_2(t))$	$F_1(\omega) F_2(\omega)$
Convolution in Frequency (Section 11.2.2.6: Convolution)	$f_1(t) f_2(t)$	$\frac{1}{2\pi} (F_1(\omega), F_2(\omega))$
Differentiation (Section 11.2.2.7: Time Differentiation)	$\frac{d^n}{dt^n} f(t)$	$(j\omega)^n F(\omega)$

11.2.2 Discussion of Fourier Transform Properties

After glancing at the above table and getting a feel for the properties of the CTFT, we will now take a little more time to discuss some of the more interesting, and more useful, properties.

11.2.2.1 Linearity

The combined addition and scalar multiplication properties in the table above demonstrate the basic property of linearity. What you should see is that if one takes the Fourier transform of a linear combination of signals then it will be the same as the linear combination of the Fourier transforms of each of the individual signals. This is crucial when using a table (Section 10.3) of transforms to find the transform of a more complicated signal.

Example 11.1

We will begin with the following signal:

$$z(t) = \alpha f_1(t) + \alpha f_2(t) \quad (11.5)$$

Now, after we take the Fourier transform, shown in the equation below, notice that the linear combination of the terms is unaffected by the transform.

$$Z(\omega) = \alpha F_1(\omega) + \alpha F_2(\omega) \quad (11.6)$$

11.2.2.2 Symmetry

Symmetry is a property that can make life quite easy when solving problems involving Fourier transforms. Basically what this property says is that since a rectangular function in time is a sinc function in frequency, then a sinc function in time will be a rectangular function in frequency. This is a direct result of the similarity between the forward CTFT and the inverse CTFT. The only difference is the scaling by 2π and a frequency reversal.

11.2.2.3 Time Scaling

This property deals with the effect on the frequency-domain representation of a signal if the time variable is altered. The most important concept to understand for the time scaling property is that signals that are narrow in time will be broad in frequency and *vice versa*. The simplest example of this is a delta function, a unit pulse⁶ with a very small duration, in time that becomes an infinite-length constant function in frequency.

The table above shows this idea for the general transformation from the time-domain to the frequency-domain of a signal. You should be able to easily notice that these equations show the relationship mentioned previously: if the time variable is increased then the frequency range will be decreased.

11.2.2.4 Time Shifting

Time shifting shows that a shift in time is equivalent to a linear phase shift in frequency. Since the frequency content depends only on the shape of a signal, which is unchanged in a time shift, then only the phase spectrum will be altered. This property can be easily proved using the Fourier Transform, so we will show the basic steps below:

Example 11.2

We will begin by letting $z(t) = f(t - \tau)$. Now let us take the Fourier transform with the previous expression substituted in for $z(t)$.

$$Z(\omega) = \int_{-\infty}^{\infty} f(t - \tau) e^{-j\omega t} dt \quad (11.7)$$

Now let us make a simple change of variables, where $\sigma = t - \tau$. Through the calculations below, you can see that only the variable in the exponential are altered thus only changing the phase in the frequency domain.

$$\begin{aligned} Z(\omega) &= \int_{-\infty}^{\infty} f(\sigma) e^{-j\omega(\sigma+\tau)} d\sigma \\ &= e^{-j\omega\tau} \int_{-\infty}^{\infty} f(\sigma) e^{-j\omega\sigma} d\sigma \\ &= e^{-j\omega\tau} F(\omega) \end{aligned} \quad (11.8)$$

11.2.2.5 Modulation (Frequency Shift)

Modulation is absolutely imperative to communications applications. Being able to shift a signal to a different frequency, allows us to take advantage of different parts of the electromagnetic spectrum is what allows us to transmit television, radio and other applications through the same space without significant interference.

The proof of the frequency shift property is very similar to that of the time shift (Section 11.2.2.4: Time Shifting); however, here we would use the inverse Fourier transform in place of the Fourier transform. Since

⁶"Elemental Signals": Section Pulse <<http://cnx.org/content/m0004/latest/#pulsedef>>

we went through the steps in the previous, time-shift proof, below we will just show the initial and final step to this proof:

$$z(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega - \phi) e^{j\omega t} d\omega \quad (11.9)$$

Now we would simply reduce this equation through another change of variables and simplify the terms. Then we will prove the property expressed in the table above:

$$z(t) = f(t) e^{j\phi t} \quad (11.10)$$

11.2.2.6 Convolution

Convolution is one of the big reasons for converting signals to the frequency domain, since convolution in time becomes multiplication in frequency. This property is also another excellent example of symmetry between time and frequency. It also shows that there may be little to gain by changing to the frequency domain when multiplication in time is involved.

We will introduce the convolution integral here, but if you have not seen this before or need to refresh your memory, then look at the continuous-time convolution (Section 3.2) module for a more in depth explanation and derivation.

$$\begin{aligned} y(t) &= (f_1(t), f_2(t)) \\ &= \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau \end{aligned} \quad (11.11)$$

11.2.2.7 Time Differentiation

Since LTI (Section 2.1) systems can be represented in terms of differential equations, it is apparent with this property that converting to the frequency domain may allow us to convert these complicated differential equations to simpler equations involving multiplication and addition. This is often looked at in more detail during the study of the Laplace Transform (Section 13.1).

Solutions to Exercises in Chapter 11

Solution to Exercise 11.1 (p. 192)

In order to calculate the Fourier transform, all we need to use is (11.1) (Continuous-Time Fourier Transform), complex exponentials (Section 1.6), and basic calculus.

$$\begin{aligned}
 \mathcal{F}(\Omega) &= \int_{-\infty}^{\infty} f(t) e^{-j\Omega t} dt \\
 &= \int_0^{\infty} e^{-\alpha t} e^{-j\Omega t} dt \\
 &= \int_0^{\infty} e^{(-t)(\alpha + j\Omega)} dt \\
 &= 0 - \frac{-1}{\alpha + j\Omega}
 \end{aligned} \tag{11.12}$$

$$\mathcal{F}(\Omega) = \frac{1}{\alpha + j\Omega} \tag{11.13}$$

Solution to Exercise 11.2 (p. 192)

Here we will use (11.2) (Inverse CTFT) to find the inverse FT given that $t \neq 0$.

$$\begin{aligned}
 x(t) &= \frac{1}{2\pi} \int_{-M}^M e^{j\Omega t} d\Omega \\
 &= \frac{1}{2\pi} e^{j\Omega t} \Big|_{\Omega=-M}^{\Omega=M} \\
 &= \frac{1}{\pi t} \sin(Mt)
 \end{aligned} \tag{11.14}$$

$$x(t) = \frac{M}{\pi} \left(\text{sinc} \frac{Mt}{\pi} \right) \tag{11.15}$$

Chapter 12

Sampling Theorem

12.1 Sampling¹

12.1.1 Introduction

The digital computer can process **discrete time signals** using extremely flexible and powerful algorithms. However, most signals of interest are **continuous time**, which is how they almost always appear in nature.

This module introduces the idea of translating continuous time problems into discrete time, and you can read on to learn more of the details and importance of **sampling**.

Key Questions

- How do we turn a continuous time signal into a discrete time signal (sampling, A/D)?
- When can we reconstruct (Section 12.2) a CT signal exactly from its samples (reconstruction, D/A)?
- Manipulating the DT signal does what to the reconstructed signal?

12.1.2 Sampling

Sampling (and reconstruction) are best understood in the frequency domain. We'll start by looking at some examples

Exercise 12.1

(Solution on p. 217.)

What CT signal $f(t)$ has the CTFT (Section 11.1) shown below?

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) e^{j\omega t} d\omega$$

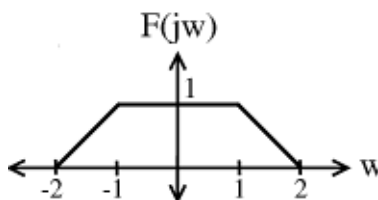


Figure 12.1: The CTFT of $f(t)$.

¹This content is available online at <<http://cnx.org/content/m10798/2.6/>>.

Hint: $F(j\omega) = F_1(j\omega) * F_2(j\omega)$ where the two parts of $F(j\omega)$ are:

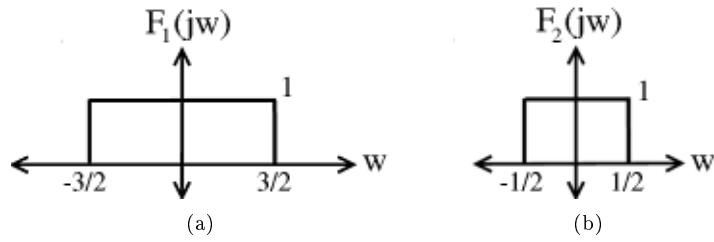


Figure 12.2

Exercise 12.2

What DT signal $f_s[n]$ has the DTFT (Section 10.4) shown below?

(Solution on p. 217.)

$$f_s[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} f_s(e^{j\omega}) e^{j\omega n} d\omega$$

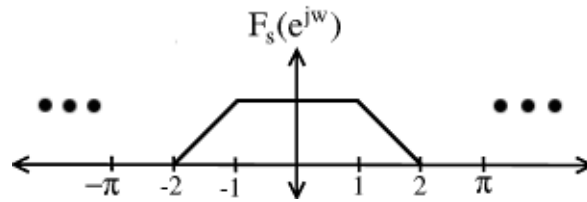


Figure 12.3: DTFT that is a periodic (with *period* = 2π) version of $F(j\omega)$ in Figure 12.1.

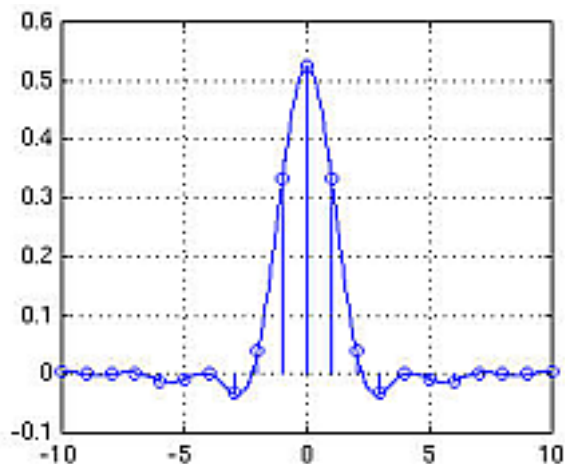


Figure 12.4: $f(t)$ is the continuous-time signal above and $f_s[n]$ is the discrete-time, sampled version of $f(t)$

12.1.2.1 Generalization

Of course, the results from the above examples can be generalized to *any* $f(t)$ with $F(j\omega) = 0$, $|\omega| > \pi$, where $f(t)$ is **bandlimited** to $[-\pi, \pi]$.

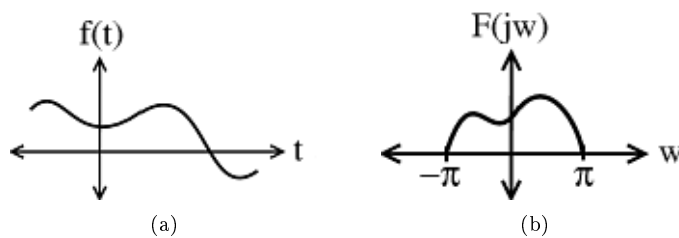


Figure 12.5: $F(j\omega)$ is the CTFT of $f(t)$.

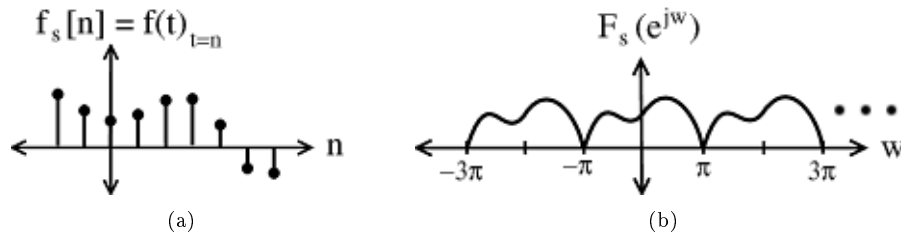


Figure 12.6: $F_s(e^{j\omega})$ is the DTFT of $f_s[n]$.

$F_s(e^{j\omega})$ is a periodic (Section 6.1) (with period 2π) version of $F(j\omega)$. $F_s(e^{j\omega})$ is the DTFT of signal sampled at the integers. $F(j\omega)$ is the CTFT of signal.

CONCLUSION: If $f(t)$ is bandlimited to $[-\pi, \pi]$ then the DTFT of the sampled version

$$f_s[n] = f(n)$$

is just a periodic (with period 2π) version of $F(j\omega)$.

12.1.3 Turning a Discrete Signal into a Continuous Signal

Now, let's look at turning a DT signal into a continuous time signal. Let $f_s[n]$ be a DT signal with DTFT $F_s(e^{j\omega})$

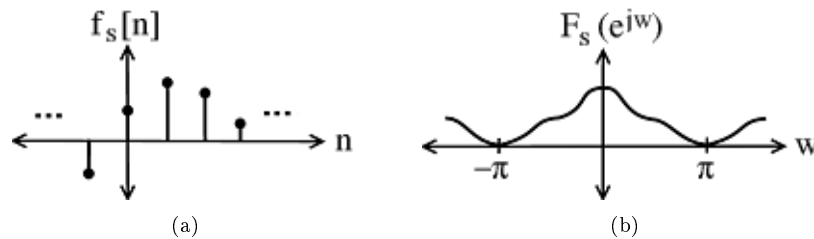


Figure 12.7: $F_s(e^{j\omega})$ is the DTFT of $f_s[n]$.

Now, set

$$f_{imp}(t) = \sum_{n=-\infty}^{\infty} (f_s[n] \delta(t - n))$$

The CT signal, $f_{imp}(t)$, is non-zero only on the integers where there are impulses of height $f_s[n]$.

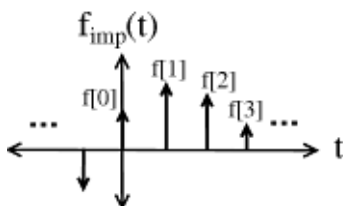


Figure 12.8

Exercise 12.3*(Solution on p. 217.)*

What is the CTFT of $f_{imp}(t)$?

Now, given the samples $f_s[n]$ of a bandlimited to $[-\pi, \pi]$ signal, our next step will be to see how we can reconstruct (Section 12.2) $f(t)$.

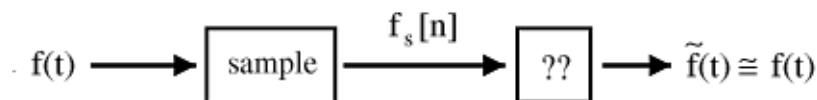


Figure 12.9: Block diagram showing the very basic steps used to reconstruct $f(t)$. Can we make our results equal $f(t)$ exactly?

12.2 Reconstruction²

12.2.1 Introduction

The reconstruction process begins by taking a sampled signal, which will be in discrete time, and performing a few operations in order to convert them into continuous-time and, with any luck, into an exact copy of the original signal. A basic method used to reconstruct a $[-\pi, \pi]$ bandlimited signal from its samples on the integer is to do the following steps:

- turn the sample sequence $f_s[n]$ into an impulse train $f_{imp}(t)$
- lowpass filter $f_{imp}(t)$ to get the reconstruction $\tilde{f}(t)$ (cutoff freq. = π)

²This content is available online at <http://cnx.org/content/m10788/2.5/>.

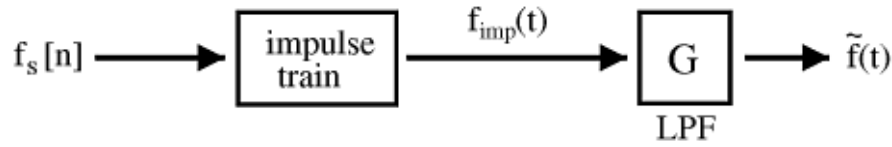


Figure 12.10: Reconstruction block diagram with lowpass filter (LPF).

The lowpass filter's impulse response is $g(t)$. The following equations allow us to reconstruct our signal (Figure 12.11), $\tilde{f}(t)$.

$$\begin{aligned}
 \tilde{f}(t) &= g(t) f_{imp}(t) \\
 &= g(t) \sum_{n=-\infty}^{\infty} (f_s[n] \delta(t-n)) \\
 &= \tilde{f}(t) \\
 &= \sum_{n=-\infty}^{\infty} (f_s[n] (g(t) \delta(t-n))) \\
 &= \sum_{n=-\infty}^{\infty} (f_s[n] g(t-n))
 \end{aligned} \tag{12.1}$$

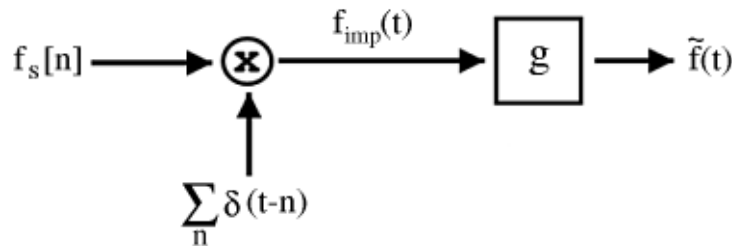


Figure 12.11

12.2.1.1 Examples of Filters g

Example 12.1: Zero Order Hold

This type "filter" is one of the most basic types of reconstruction filters. It simply holds the value that is in $f_s[n]$ for τ seconds. This creates a block or step like function where each value of the pulse in $f_s[n]$ is simply dragged over to the next pulse. The equations and illustrations below (Figure 12.12) depict how this reconstruction filter works with the following g :

$$g(t) = \begin{cases} 1 & \text{if } 0 < t < \tau \\ 0 & \text{otherwise} \end{cases}$$

$$f_s[n] = \sum_{n=-\infty}^{\infty} (f_s[n] g(t-n)) \tag{12.2}$$

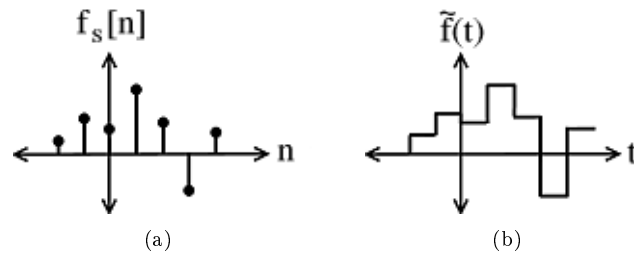


Figure 12.12: Zero Order Hold

QUESTION: How does $\tilde{f}(t)$ reconstructed with a zero order hold compare to the original $f(t)$ in the frequency domain?

Example 12.2: Nth Order Hold

Here we will look at a few quick examples of variances to the Zero Order Hold filter discussed in the previous example.

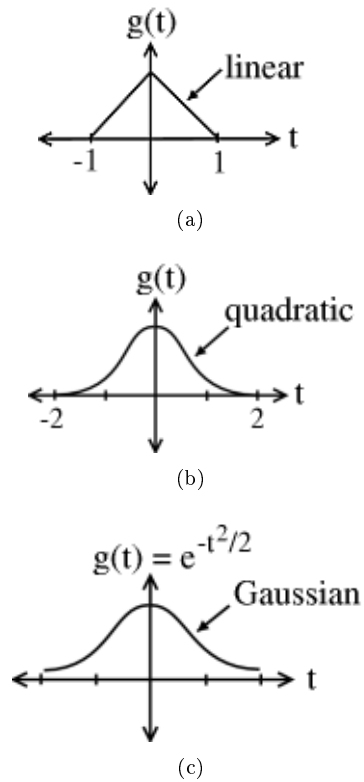


Figure 12.13: Nth Order Hold Examples (nth order hold is equal to an nth order B-spline) (a) First Order Hold (b) Second Order Hold (c) ∞ Order Hold

12.2.2 Ultimate Reconstruction Filter

QUESTION: What is the ultimate reconstruction filter?

Recall that (see Figure 12.14)

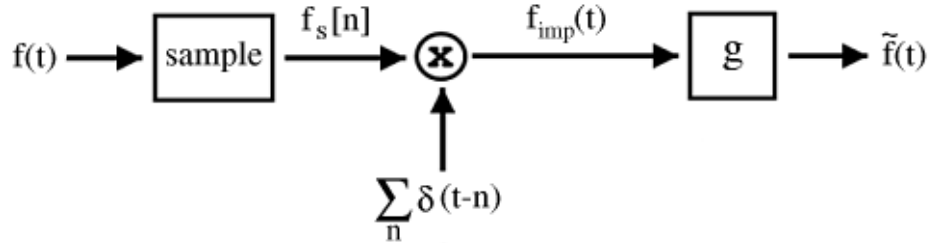


Figure 12.14: Our current reconstruction block diagram. Note that each of these signals has its own corresponding CTFT or DTFT.

If $G(j\omega)$ has the following shape (Figure 12.15):

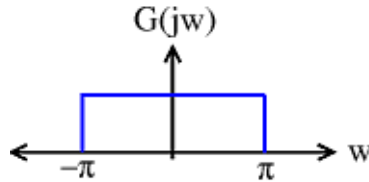


Figure 12.15: Ideal lowpass filter

then

$$\tilde{f}(t) = f(t)$$

Therefore, an ideal lowpass filter will give us perfect reconstruction!

In the time domain, impulse response

$$g(t) = \frac{\sin(\pi t)}{\pi t} \quad (12.3)$$

$$\begin{aligned} \tilde{f}(t) &= \sum_{n=-\infty}^{\infty} (f_s[n] g(t-n)) \\ &= \sum_{n=-\infty}^{\infty} \left(f_s[n] \frac{\sin(\pi(t-n))}{\pi(t-n)} \right) \\ &= f(t) \end{aligned} \quad (12.4)$$

12.2.3 Amazing Conclusions

If $f(t)$ is bandlimited to $[-\pi, \pi]$, it can be reconstructed perfectly from its samples on the integers $f_s[n] = f(t)|_{t=n}$

$$f(t) = \sum_{n=-\infty}^{\infty} \left(f_s[n] \frac{\sin(\pi(t-n))}{\pi(t-n)} \right) \quad (12.5)$$

The above equation for perfect reconstruction deserves a closer look (Section 12.3), which you should continue to read in the following section to get a better understanding of reconstruction. Here are a few things to think about for now:

- What does $\frac{\sin(\pi(t-n))}{\pi(t-n)}$ equal at integers other than n ?
- What is the support of $\frac{\sin(\pi(t-n))}{\pi(t-n)}$?

12.3 More on Reconstruction³

12.3.1 Introduction

In the previous module on reconstruction (Section 12.2), we gave an introduction into how reconstruction works and briefly derived an equation used to perform perfect reconstruction. Let us now take a closer look at the perfect reconstruction formula:

$$f(t) = \sum_{n=-\infty}^{\infty} \left(f_s \frac{\sin(\pi(t-n))}{\pi(t-n)} \right) \quad (12.6)$$

We are writing $f(t)$ in terms of shifted and scaled sinc functions.

$$\left\{ \frac{\sin(\pi(t-n))}{\pi(t-n)} \right\}_{n \in \mathbb{Z}}$$

is a **basis** (Section 5.1.3: Basis) for the space of $[-\pi, \pi]$ bandlimited signals. But wait

12.3.1.1 Derive Reconstruction Formulas

What is

$$\left\langle \frac{\sin(\pi(t-n))}{\pi(t-n)}, \frac{\sin(\pi(t-k))}{\pi(t-k)} \right\rangle =? \quad (12.7)$$

This inner product (Section 15.3) can be hard to calculate in the time domain, so let's use Plancherel Theorem (Section 15.12)

$$\langle \cdot, \cdot \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-(j\omega n)} e^{j\omega k} d\omega \quad (12.8)$$

³This content is available online at <http://cnx.org/content/m10790/2.4/>.

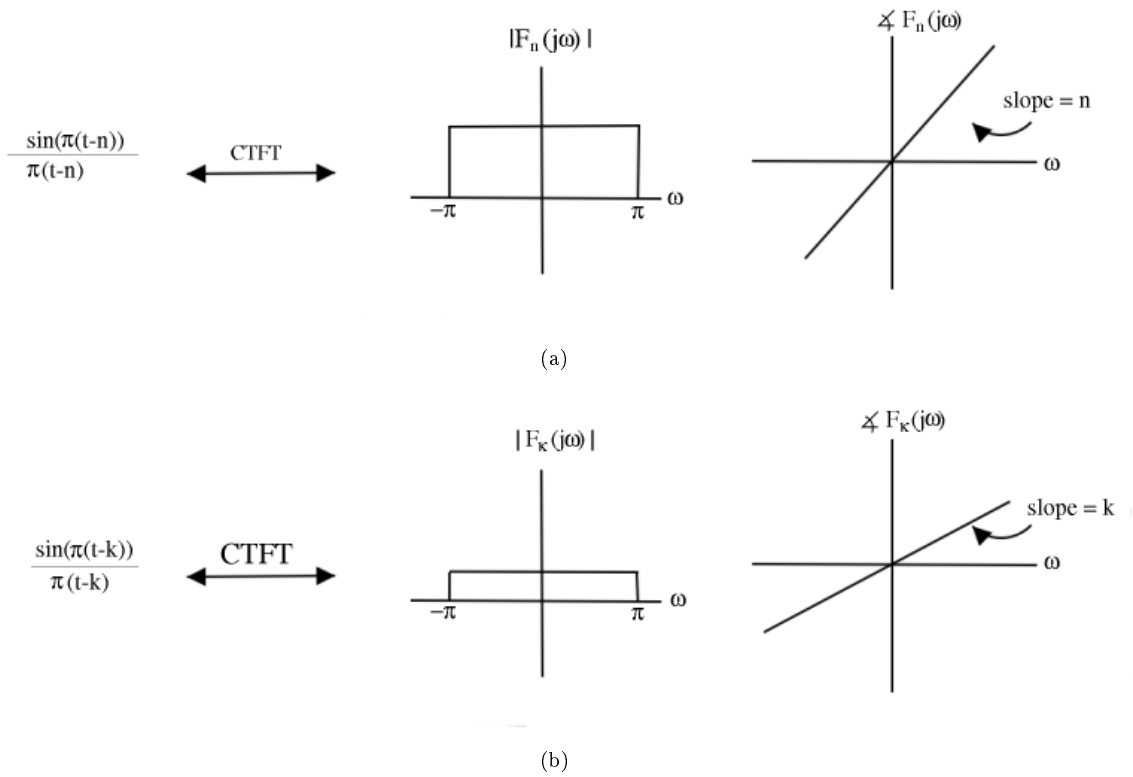


Figure 12.16

if $n = k$

$$\begin{aligned} \langle \text{sinc}_n, \text{sinc}_k \rangle &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-j\omega n} e^{j\omega k} d\omega \\ &= 1 \end{aligned} \quad (12.9)$$

if $n \neq k$

$$\begin{aligned} \langle \text{sinc}_n, \text{sinc}_k \rangle &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-j\omega n} e^{j\omega k} d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega(k-n)} d\omega \\ &= \frac{1}{2\pi} \frac{\sin(\pi(k-n))}{j(k-n)} \\ &= 0 \end{aligned} \quad (12.10)$$

NOTE: In (12.10) we used the fact that the integral of sinusoid over a complete interval is 0 to simplify our equation.

So,

$$\left\langle \frac{\sin(\pi(t-n))}{\pi(t-n)}, \frac{\sin(\pi(t-k))}{\pi(t-k)} \right\rangle = \begin{cases} 1 & \text{if } n = k \\ 0 & \text{if } n \neq k \end{cases} \quad (12.11)$$

Therefore

$$\left\{ \frac{\sin(\pi(t-n))}{\pi(t-n)} \right\}_{n \in \mathbb{Z}}$$

is an orthonormal basis (Section 15.7.3: Orthonormal Basis) (ONB) for the space of $[-\pi, \pi]$ bandlimited functions.

SAMPLING: Sampling is the same as calculating ONB coefficients, which is inner products with sines

12.3.1.2 Summary

One last time for $f(t)$ $[-\pi, \pi]$ bandlimited

Synthesis

$$f(t) = \sum_{n=-\infty}^{\infty} \left(f_s[n] \frac{\sin(\pi(t-n))}{\pi(t-n)} \right) \quad (12.12)$$

Analysis

$$f_s[n] = f(t)|_{t=n} \quad (12.13)$$

In order to understand a little more about how we can reconstruct a signal exactly, it will be useful to examine the relationships⁴ between the fourier transforms (CTFT and DTFT) in more depth.

12.4 Nyquist Theorem⁵

12.4.1 Introduction

Earlier you should have been exposed to the concepts behind sampling (Section 12.1) and the sampling theorem. While learning about these ideas, you should have begun to notice that if we sample at too low of a rate, there is a chance that our original signal will not be uniquely defined by our sampled signal. If this happens, then there is no guarantee that we can correctly reconstruct (Section 12.2) the signal. As a result of this, the **Nyquist Theorem** was created. Below, we will discuss just what exactly this theorem tells us.

12.4.2 Nyquist Theorem

We will let T equal our sampling period (distance between samples). Then let $\Omega_s = \frac{2\pi}{T}$ (sampling frequency in radians/sec). We have seen that if $f(t)$ is bandlimited to $[-\Omega_B, \Omega_B]$ and we sample with period $T < \frac{\pi}{\Omega_b} \Rightarrow \frac{2\pi}{\Omega_s} < \frac{\pi}{\Omega_B} \Rightarrow \Omega_s > 2\Omega_B$ then we can reconstruct $f(t)$ from its samples.

Theorem 12.1: Nyquist Theorem ("Fundamental Theorem of DSP")

If $f(t)$ is bandlimited to $[-\Omega_B, \Omega_B]$, we can reconstruct it *perfectly* from its samples

$$f_s[n] = f(nT)$$

for $\Omega_s = \frac{2\pi}{T} > 2\Omega_B$

$\Omega_N = 2\Omega_B$ is called the "**Nyquist frequency**" for $f(t)$. For perfect reconstruction to be possible

$$\Omega_s \geq 2\Omega_B$$

where Ω_s is the sampling frequency and Ω_B is the highest frequency in the signal.

⁴"Examining Reconstruction Relations" <<http://cnx.org/content/m10799/latest/>>

⁵This content is available online at <<http://cnx.org/content/m10791/2.5/>>.

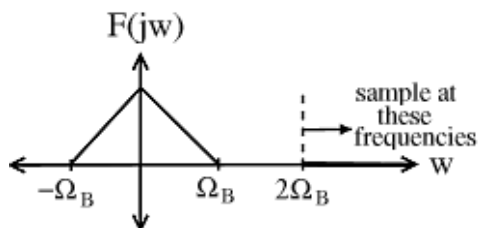


Figure 12.17: Illustration of Nyquist Frequency

Example 12.3: Examples:

- Human ear hears frequencies up to 20 kHz \rightarrow CD sample rate is 44.1 kHz.
- Phone line passes frequencies up to 4 kHz \rightarrow phone company samples at 8 kHz.

12.4.2.1 Reconstruction

The reconstruction formula in the time domain looks like

$$f(t) = \sum_{n=-\infty}^{\infty} \left(f_s[n] \frac{\sin\left(\frac{\pi}{T}(t-nT)\right)}{\frac{\pi}{T}(t-nT)} \right)$$

We can conclude, just as before, that

$$\frac{\sin\left(\frac{\pi}{T}(t-nT)\right)}{\frac{\pi}{T}(t-nT)}, \quad n \in \mathbb{Z}$$

is a basis (Section 5.1) for the space of $[-\Omega_B, \Omega_B]$ bandlimited functions, $\Omega_B = \frac{\pi}{T}$. The expansion coefficients for this basis are calculated by sampling $f(t)$ at rate $\frac{2\pi}{T} = 2\Omega_B$.

NOTE: The basis is also orthogonal. To make it orthonormal (Section 15.8), we need a normalization factor of \sqrt{T} .

12.4.2.2 The Big Question

Exercise 12.4

(Solution on p. 217.)

What if $\Omega_s < 2\Omega_B$? What happens when we sample below the Nyquist rate?

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10791/latest/NyquistPlot.llb>

12.5 Aliasing⁶

12.5.1 Introduction

When considering the reconstruction (Section 12.2) of a signal, you should already be familiar with the idea of the Nyquist rate (Section 12.4). This concept allows us to find the sampling rate that will provide for perfect reconstruction of our signal. If we sample at too low of a rate (below the Nyquist rate), then problems will arise that will make perfect reconstruction impossible - this problem is known as **aliasing**. Aliasing occurs when there is an overlap in the shifted, periodic copies of our original signal's FT, *i.e.* spectrum.

In the frequency domain, one will notice that part of the signal will overlap with the periodic signals next to it. In this overlap the values of the frequency will be added together and the shape of the signals spectrum will be unwantingly altered. This overlapping, or aliasing, makes it impossible to correctly determine the correct strength of that frequency. Figure 12.18 provides a visual example of this phenomenon:

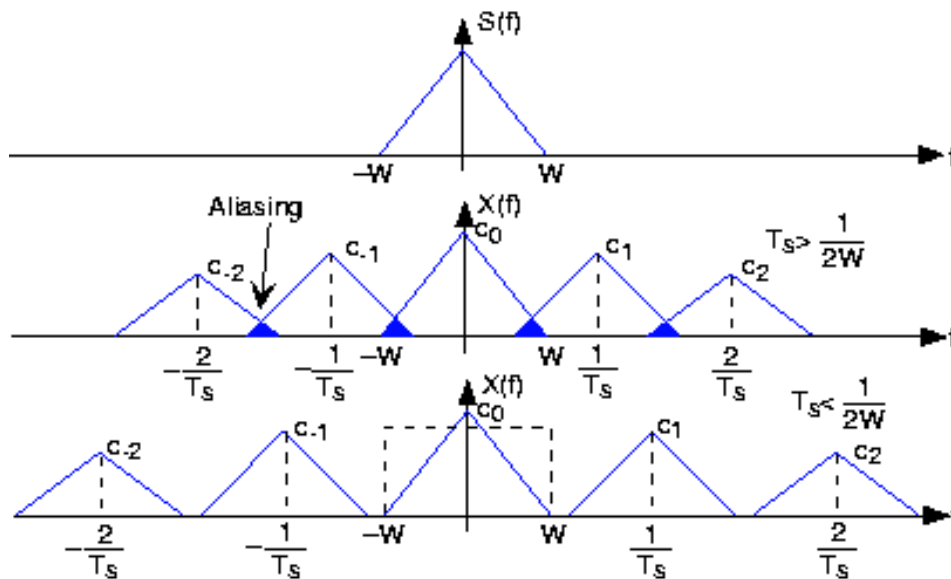


Figure 12.18: The spectrum of some bandlimited (to W Hz) signal is shown in the top plot. If the sampling interval T_s is chosen too large relative to the bandwidth W , aliasing will occur. In the bottom plot, the sampling interval is chosen sufficiently small to avoid aliasing. Note that if the signal were not bandlimited, the component spectra would always overlap.

12.5.2 Aliasing and Sampling

If we sample too slowly, *i.e.*,

$$\Omega_s < 2\Omega_B \quad , \quad T > \frac{\pi}{\Omega_B}$$

We cannot recover the signal from its samples due to aliasing.

Example 12.4

⁶This content is available online at <http://cnx.org/content/m10793/2.6/>.

Let $f_1(t)$ have CTFT.

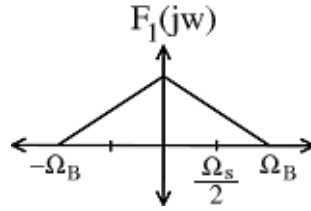


Figure 12.19: In this figure, note the following equation: $\Omega_B - \frac{\Omega_s}{2} = a$

Let $f_2(t)$ have CTFT.

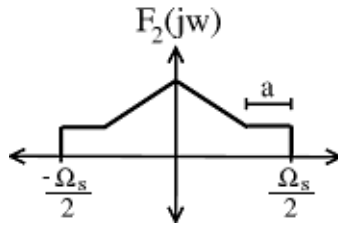


Figure 12.20: The horizontal portions of the signal result from overlap with shifted replicas - showing visual proof of aliasing.

Try to sketch and answer the following questions on your own:

- What does the DTFT of $f_{1,s}[n] = f_1(nT)$ look like?
- What does the DTFT of $f_{2,s}[n] = f_2(nT)$ look like?
- Do any other signals have the same DTFT as $f_{1,s}[n]$ and $f_{2,s}[n]$?

CONCLUSION: If we sample below the Nyquist frequency, there are many signals that could have produced that given sample sequence.

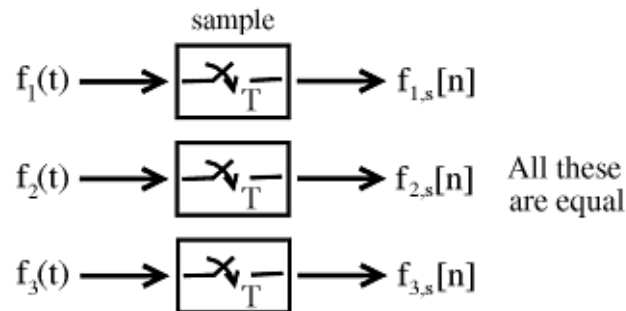


Figure 12.21: These are all equal!

Why the term "aliasing"? Because the same sample sequence can represent different CT signals (as opposed to when we sample above the Nyquist frequency, then the sample sequence represents a unique CT signal).

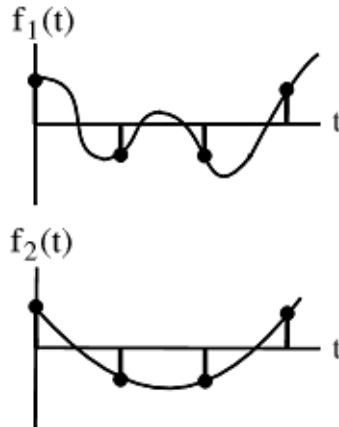


Figure 12.22: These two signals contain the same four samples, yet are very different signals.

Example 12.5

$$f(t) = \cos(2\pi t)$$

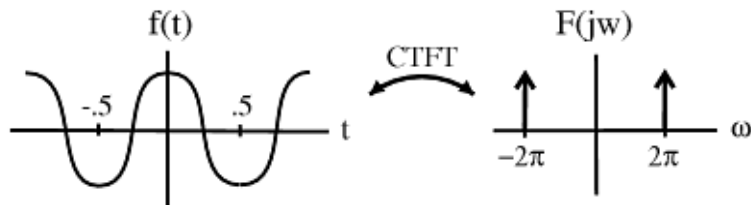


Figure 12.23: The cosine function, $f(t) = \cos(2\pi t)$, and its CTFT.

Case 1: Sample $\Omega_s = (8\pi) \frac{\text{rad}}{\text{sec}} \Rightarrow T = \frac{1}{4} \text{sec}$.

NOTE: $\Omega_s > 2\Omega_B$

Case 2: Sample $\frac{\omega}{\Omega_s} = (\frac{8}{3}\pi) \frac{\text{rad}}{\text{sec}} \Rightarrow T = \frac{3}{4} \text{sec}$.

NOTE: $\Omega_s < 2\Omega_B$

When we run the DTFT from Case #2 through the reconstruction steps, we realize that we end up with the following cosine:

$$\tilde{f}(t) = \cos\left(\frac{\pi}{2}t\right)$$

This is a "stretched" out version of our original. Clearly, our sampling rate was not high enough to ensure correct reconstruction from the samples.

You may have seen some effects of aliasing such as a wagon wheel turning backwards in a western movie. Aliasing in images⁷ can result in Moire Patterns. Here is an example of an image that has Moire artifacts⁸ as a result of scanning at too low a frequency.

This is an unsupported media type. To view, please see <http://cnx.org/content/m10793/latest/alias.llb>

12.6 Anti-Aliasing Filters⁹

12.6.1 Introduction

The idea of aliasing (Section 12.5) has been described as the problem that occurs if a signal is not sampled (Section 12.1) at a high enough rate (for example, below the Nyquist Frequency (Section 12.4)). But exactly what kind of distortion does aliasing produce?

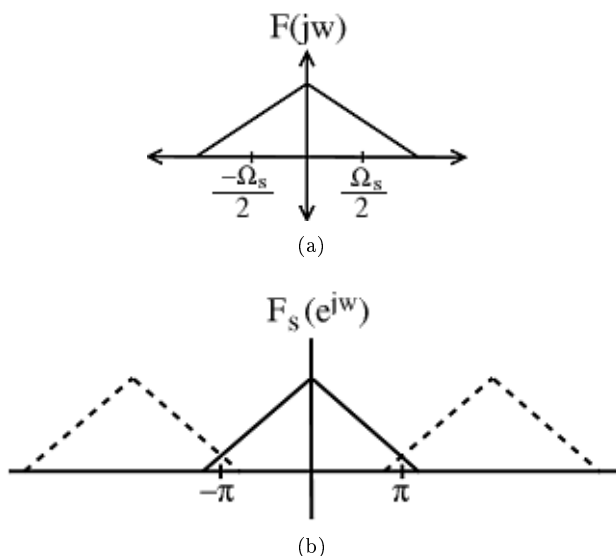


Figure 12.24

High frequencies in the original signal "fold back" into lower frequencies.

High frequencies masquerading as lower frequencies produces *highly undesirable* artifacts in the reconstructed signal.

WARNING: We must avoid aliasing anyway we can.

⁷<http://ptolemy.eecs.berkeley.edu/eecs20/week13/moire.html>

⁸<http://www.dvp.co.il/filter/moire.html>

⁹This content is available online at <http://cnx.org/content/m10794/2.4/>.

12.6.2 Avoiding Aliasing

What if it is impractical/impossible to sample at $\Omega_s > 2\Omega_B$?

Filter out the frequencies above $\frac{\Omega_s}{2}$ before you sample. The best way to visualize doing this is to imagine the following simple steps:

1. Take the CTFT of the signal, $f(t)$.
2. Send this signal through a lowpass filter with the following specification, $\omega_c = \frac{\Omega_s}{2}$.
3. We now have a graph of our signal in the frequency domain with all values of $|\omega| > \frac{\Omega_s}{2}$ equal to zero. Now, we take the inverse CTFT to get back our continuous time signal, $f_a(t)$.
4. And finally we are ready to sample our signal!

Example 12.6

Sample rate for *CD* = 44.1KHz.

Many musical instruments (e.g. highhat) contain frequencies above 22KHz (even though we cannot hear them).

Because of this, we can filter the output signal from the instrument before we sample it using the following filter:

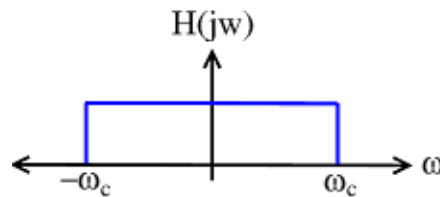


Figure 12.25: This filter will cutoff the higher, unnecessary frequencies, where $|\omega_c| > 2\pi 22kHz$

Now the signal is ready to be sampled!

Example 12.7: Another Example

Speech bandwidth is $> \pm(20kHz)$, but it is perfectly intelligible when lowpass filtered to a $\pm(4kHz)$ range. Because of this, we can take a normal speech signal and pass it through a filter like the one shown in Figure 12.25, where we now set $|\omega_c| > 2\pi 4kHz$. The signal we receive from this filter only contains values where $|\omega| > 8\pi k$.

Now we can sample at $16\pi k = 8kHz$ – standard telephony rate.

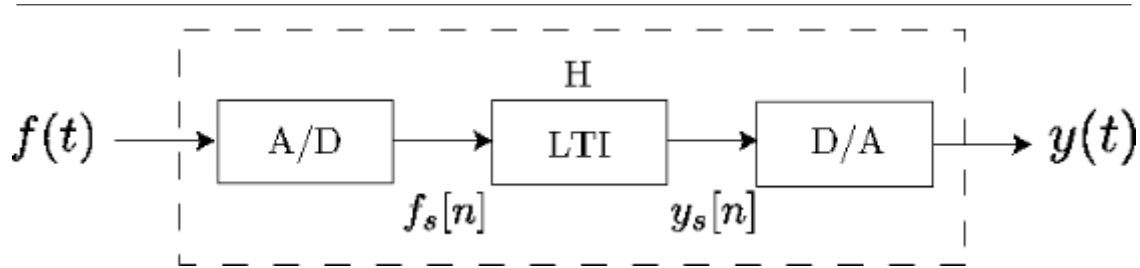
12.7 Discrete Time Processing of Continuous Time Signals¹⁰

Figure 12.26: DSP System

How is the CTFT of $y(t)$ related to the CTFT of $f(t)$ (Figure 1)?

Let $G(j\omega)$ = reconstruction filter freq. response

$$Y(j\omega) = G(j\omega) Y_{imp}(j\omega)$$

where $Y_{imp}(j\omega)$ is impulse sequence created from $y_s[n]$. So,

$$Y(j\omega) = G(j\omega) Y_s(e^{j\omega T}) = G(j\omega) H(e^{j\omega T}) F_s(e^{j\omega T})$$

$$Y(j\omega) = G(j\omega) H(e^{j\omega T}) \left(\frac{1}{T} \sum_{r=-\infty}^{\infty} \left(F\left(j\frac{\omega F 2\pi r}{T}\right) \right) \right)$$

$$Y(j\omega) = \frac{1}{T} G(j\omega) H(e^{j\omega T}) \sum_{r=-\infty}^{\infty} \left(F\left(j\frac{\omega F 2\pi r}{T}\right) \right)$$

Now, lets assume that $f(t)$ is bandlimited to $[-(\frac{\pi}{T}), \frac{\pi}{T}] = [-(\frac{\Omega_s}{2}), \frac{\Omega_s}{2}]$ and $G(j\omega)$ is a perfect reconstruction filter. Then

$$Y(j\omega) = \begin{cases} F(j\omega) H(e^{j\omega T}) & \text{if } |\omega| \leq \frac{\pi}{T} \\ 0 & \text{otherwise} \end{cases}$$

NOTE: $Y(j\omega)$ has the same "bandlimit" as $F(j\omega)$.

So, for bandlimited signals, and with a high enough sampling rate and a perfect reconstruction filter (Figure 2)

¹⁰This content is available online at <<http://cnx.org/content/m10797/2.9/>>.

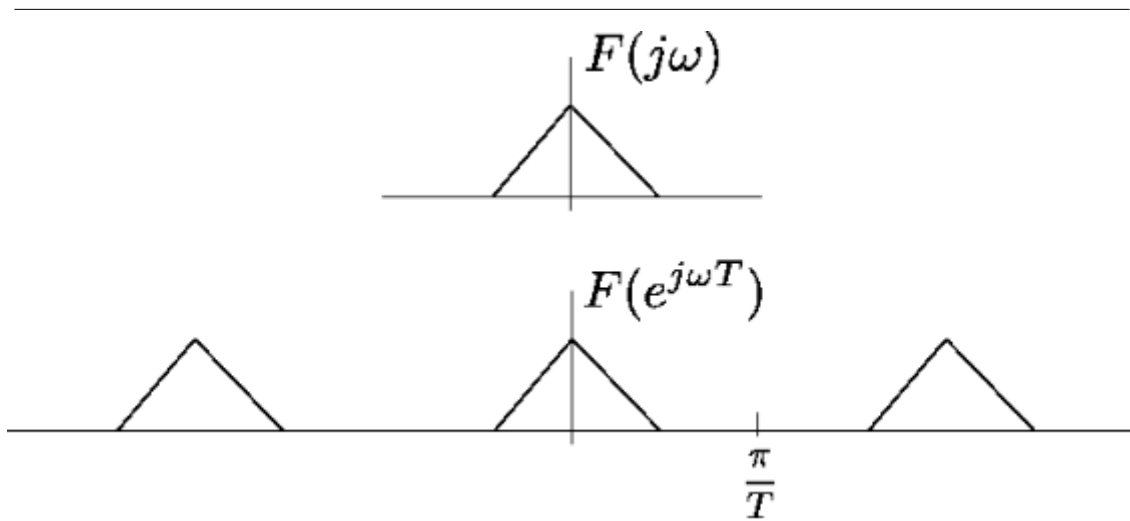


Figure 12.27: FT's of original (analog) signal $f(t)$ and sampled version of $f(t)$ respectively.

is equivalent to using an analog LTI filter (Figure 3)

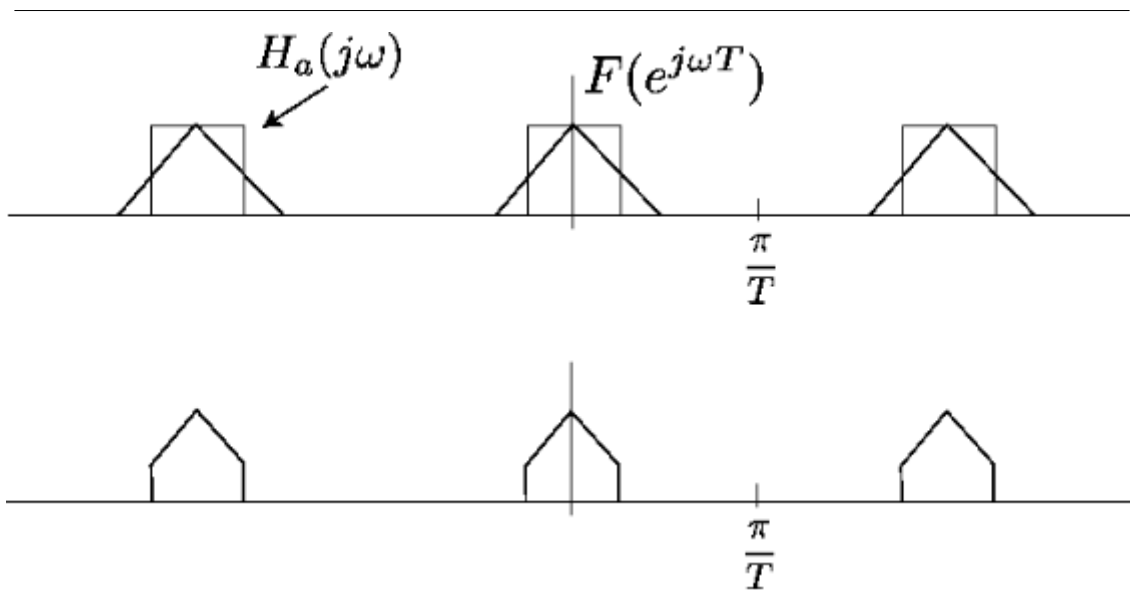


Figure 12.28: Implementing a discrete time filter (H) in analog

where

$$H_a(j\omega) = \begin{cases} H(e^{j\omega T}) & \text{if } |\omega| \leq \frac{\pi}{T} \\ 0 & \text{otherwise} \end{cases}$$

So, by being careful we can implement LTI systems for bandlimited signals **on our computer!!!**

Important note:

$H_a(j\omega)$ = filter induced by our system.

$H_a(j\omega)$ is LTI only if

- h , the DT system, is LTI
- $F(j\omega)$, the input, is bandlimited and the sample rate is high enough.

Solutions to Exercises in Chapter 12

Solution to Exercise 12.1 (p. 197)

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(jw) e^{jwt} dw$$

Solution to Exercise 12.2 (p. 198)

Since $F(jw) = 0$ outside of $[-2, 2]$

$$f(t) = \frac{1}{2\pi} \int_{-2}^2 F(jw) e^{jwt} dw$$

Also, since we only use one interval to reconstruct $f_s[n]$ from its DTFT, we have

$$f_s[n] = \frac{1}{2\pi} \int_{-2}^2 f_s(e^{jw}) e^{jwn} dw$$

Since $F(jw) = F_s(e^{jw})$ on $[-2, 2]$

$$f_s[n] = f(t)|_{t=n}$$

i.e. $f_s[n]$ is a *sampled* version of $f(t)$.

Solution to Exercise 12.3 (p. 201)

$$f_{imp}(t) = \sum_{n=-\infty}^{\infty} (f_s[n] \delta(t-n))$$

$$\begin{aligned} \tilde{F}_{imp}(jw) &= \int_{-\infty}^{\infty} f_{imp}(t) e^{-jw t} dt \\ &= \int_{-\infty}^{\infty} (\sum_{n=-\infty}^{\infty} (f_s[n] \delta(t-n))) e^{-jw t} dt \\ &= (\sum_{n=-\infty}^{\infty} (f_s[n])) \int_{-\infty}^{\infty} \delta(t-n) e^{-jw t} dt \\ &= (\sum_{n=-\infty}^{\infty} (f_s[n])) e^{-jwn} \\ &= F_s(e^{jw}) \end{aligned} \tag{12.14}$$

So, the CTFT of $f_{imp}(t)$ is *equal* to the DTFT of $f_s[n]$

NOTE: We used the sifting property to show $\int_{-\infty}^{\infty} \delta(t-n) e^{-jw t} dt = e^{-jwn}$

Solution to Exercise 12.4 (p. 208)

Go through the steps: (see Figure 12.29)

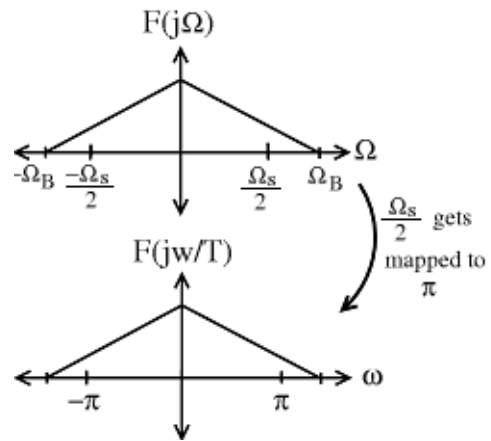


Figure 12.29

Finally, what will happen to $F_s(e^{j\omega})$ now? To answer this final question, we will now need to look into the concept of aliasing (Section 12.5).

Chapter 13

Laplace Transform and System Design

13.1 The Laplace Transforms¹

The Laplace transform is a generalization of the Continuous-Time Fourier Transform (Section 11.1). However, instead of using complex sinusoids (Section 7.2) of the form $e^{j\omega t}$, as the CTFT does, the Laplace transform uses the more general, e^{st} , where $s = \sigma + j\omega$.

Although Laplace transforms are rarely solved using integration (tables (Section 13.3) and computers (e.g. Matlab) are much more common), we will provide the **bilateral Laplace transform pair** here. These define the forward and inverse Laplace transformations. Notice the similarities between the forward and inverse transforms. This will give rise to many of the same symmetries found in Fourier analysis (Section 7.1).

Laplace Transform

$$F(s) = \int_{-\infty}^{\infty} f(t) e^{-(st)} dt \quad (13.1)$$

Inverse Laplace Transform

$$f(t) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s) e^{st} ds \quad (13.2)$$

13.1.1 Finding the Laplace and Inverse Laplace Transforms

13.1.1.1 Solving the Integral

Probably the most difficult and least used method for finding the Laplace transform of a signal is solving the integral. Although it is technically possible, it is extremely time consuming. Given how easy the next two methods are for finding it, we will not provide any more than this. The integrals are primarily there in order to understand where the following methods originate from.

13.1.1.2 Using a Computer

Using a computer to find Laplace transforms is relatively painless. Matlab has two functions, `laplace` and `ilaplace`, that are both part of the symbolic toolbox, and will find the Laplace and inverse Laplace transforms respectively. This method is generally preferred for more complicated functions. Simpler and more contrived functions are usually found easily enough by using tables (Section 13.1.1.3: Using Tables).

¹This content is available online at <<http://cnx.org/content/m10110/2.12/>>.

13.1.1.3 Using Tables

When first learning about the Laplace transform, tables are the most common means for finding it. With enough practice, the tables themselves may become unnecessary, as the common transforms can become second nature. For the purpose of this section, we will focus on the inverse Laplace transform, since most design applications will begin in the Laplace domain and give rise to a result in the time domain. The method is as follows:

1. Write the function you wish to transform, $H(s)$, as a sum of other functions, $H(s) = \sum_{i=1}^m (H_i(s))$ where each of the H_i is known from a table (Section 13.3).
2. Invert each $H_i(s)$ to get its $h_i(t)$.
3. Sum up the $h_i(t)$ to get $h(t) = \sum_{i=1}^m (h_i(t))$

Example 13.1

Compute $h(t)$ for $H(s) = \frac{1}{s+5}$, $Re(s) > -5$

This can be solved directly from the table (Section 13.3) to be $h(t) = e^{-(5t)}$

Example 13.2

Find the time domain representation, $h(t)$, of $H(s) = \frac{25}{s+10}$, $Re(s) > -10$

To solve this, we first notice that $H(s)$ can also be written as $25\frac{1}{s+10}$. We can then go to the table (Section 13.3) to find $h(t) = 25e^{-(10t)}$

Example 13.3

We can now extend the two previous examples by finding $h(t)$ for $H(s) = \frac{1}{s+5} + \frac{25}{s+10}$, $Re(s) > -5$

To do this, we take advantage of the additive property of linearity and the three-step method described above to yield the result $h(t) = e^{-(5t)} + 25e^{-(10t)}$

For more complicated examples, it may be more difficult to break up the transfer function into parts that exist in a table. In this case, it is often necessary to use partial fraction expansion² to get the transfer function into a more usable form.

13.1.2 Visualizing the Laplace Transform

With the Fourier transform, we had a **complex-valued function** of a *purely imaginary variable*, $F(j\omega)$. This was something we could envision with two 2-dimensional plots (real and imaginary parts or magnitude and phase). However, with Laplace, we have a **complex-valued function** of a *complex variable*. In order to examine the magnitude and phase or real and imaginary parts of this function, we must examine 3-dimensional surface plots of each component.

²"Partial Fraction Expansion" <<http://cnx.org/content/m2111/latest/>>

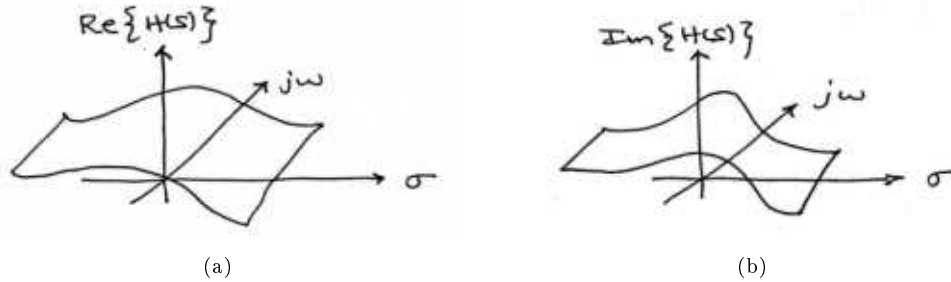
 real and imaginary sample plots


Figure 13.1: Real and imaginary parts of $H(s)$ are now each 3-dimensional surfaces. (a) The Real part of $H(s)$ (b) The Imaginary part of $H(s)$

magnitude and phase sample plots

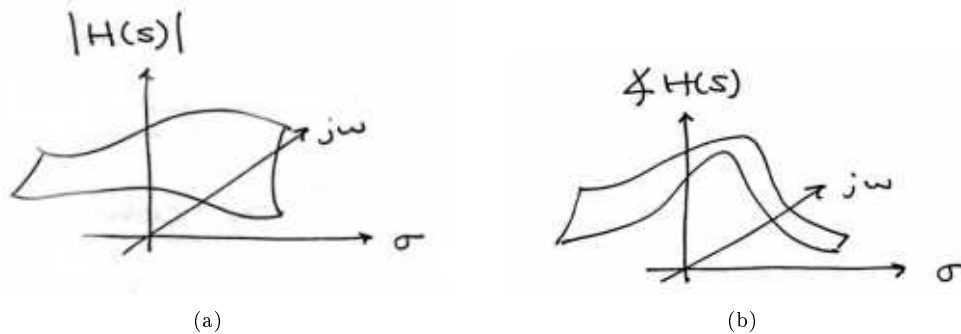


Figure 13.2: Magnitude and phase of $H(s)$ are also each 3-dimensional surfaces. This representation is more common than real and imaginary parts. (a) The Magnitude of $H(s)$ (b) The Phase of $H(s)$

While these are legitimate ways of looking at a signal in the Laplace domain, it is quite difficult to draw and/or analyze. For this reason, a simpler method has been developed. Although it will not be discussed in detail here, the method of Poles and Zeros (Section 13.6) is much easier to understand and is the way both the Laplace transform and its discrete-time counterpart the Z-transform (Section 14.1) are represented graphically.

13.2 Properties of the Laplace Transform³

Property	Signal	Laplace Transform	Region of Convergence
Linearity	$\alpha x_1(t) + \beta x_2(t)$	$\alpha X_1(s) + \beta X_2(s)$	At least $ROC_1 \cap ROC_2$
Time Shifting	$x(t - \tau)$	$e^{-(s\tau)} X(s)$	ROC
Frequency Shifting (modulation)	$e^{\eta t} x(t)$	$X(s - \eta)$	Shifted ROC ($s - \eta$ must be in the region of convergence)
Time Scaling	$x(\alpha t)$	$(1 - \alpha) X(s - \alpha)$	Scaled ROC ($s - \alpha$ must be in the region of convergence)
Conjugation	$x(t)^*$	$X(s^*)^*$	ROC
Convolution	$x_1(t) * x_2(t)$	$X_1(s) X_2(s)$	At least $ROC_1 \cap ROC_2$
Time Differentiation	$\frac{d}{dt} x(t)$	$sX(s)$	At least ROC
Frequency Differentiation	$(-t) x(t)$	$\frac{d}{ds} X(s)$	ROC
Integration in Time	$\int_{-\infty}^t x(\tau) d\tau$	$(1 - s) X(s)$	At least $ROC \cap \text{Re}(s) > 0$

³This content is available online at <<http://cnx.org/content/m10117/2.9/>>.

13.3 Table of Common Laplace Transforms⁴

Signal	Laplace Transform	Region of Convergence
$\delta(t)$	1	All s
$\delta(t - T)$	$e^{-(sT)}$	All s
$u(t)$	$\frac{1}{s}$	$Re(s) > 0$
$-(u(-t))$	$\frac{1}{s}$	$Re(s) < 0$
$tu(t)$	$\frac{1}{s^2}$	$Re(s) > 0$
$t^n u(t)$	$\frac{n!}{s^{n+1}}$	$Re(s) > 0$
$-(t^n u(-t))$	$\frac{n!}{s^{n+1}}$	$Re(s) < 0$
$e^{-(\lambda t)} u(t)$	$\frac{1}{s+\lambda}$	$Re(s) > -\lambda$
$-(e^{-(\lambda t)}) u(-t)$	$\frac{1}{s+\lambda}$	$Re(s) < -\lambda$
$te^{-(\lambda t)} u(t)$	$\frac{1}{(s-\lambda)^2}$	$Re(s) > -\lambda$
$t^n e^{-(\lambda t)} u(t)$	$\frac{n!}{(s+\lambda)^{n+1}}$	$Re(s) > -\lambda$
$-(t^n e^{-(\lambda t)} u(-t))$	$\frac{n!}{(s+\lambda)^{n+1}}$	$Re(s) < -\lambda$
$\cos(bt) u(t)$	$\frac{s}{s^2+b^2}$	$Re(s) > 0$
$\sin(bt) u(t)$	$\frac{b}{s^2+b^2}$	$Re(s) > 0$
$e^{-(at)} \cos(bt) u(t)$	$\frac{s+a}{(s+a)^2+b^2}$	$Re(s) > -a$
$e^{-(at)} \sin(bt) u(t)$	$\frac{b}{(s+a)^2+b^2}$	$Re(s) > -a$
$\frac{d^n}{dt^n} \delta(t)$	s^n	All s

13.4 Region of Convergence for the Laplace Transform⁵

With the Laplace transform (Section 13.1), the s-plane represents a set of signals (complex exponentials (Section 1.6)). For any given LTI (Section 2.1) system, some of these signals may cause the output of the system to converge, while others cause the output to diverge ("blow up"). The set of signals that cause the system's output to converge lie in the **region of convergence (ROC)**. This module will discuss how to find this region of convergence for any continuous-time, LTI system.

Recall the definition of the Laplace transform,

Laplace Transform

$$H(s) = \int_{-\infty}^{\infty} h(t) e^{-(st)} dt \quad (13.3)$$

If we consider a causal (Section 1.1), complex exponential, $h(t) = e^{-(at)} u(t)$, we get the equation,

$$\int_0^{\infty} e^{-(at)} e^{-(st)} dt = \int_0^{\infty} e^{-((a+s)t)} dt \quad (13.4)$$

Evaluating this, we get

$$\frac{-1}{s+a} \left(\lim_{t \rightarrow \infty} e^{-((s+a)t)} - 1 \right) \quad (13.5)$$

⁴This content is available online at <<http://cnx.org/content/m10111/2.10/>>.

⁵This content is available online at <<http://cnx.org/content/m10114/2.8/>>.

Notice that this equation will tend to infinity when $\lim_{t \rightarrow \infty} e^{-((s+a)t)}$ tends to infinity. To understand when this happens, we take one more step by using $s = \sigma + j\omega$ to realize this equation as

$$\lim_{t \rightarrow \infty} e^{-(j\omega t)} e^{-((\sigma+a)t)} \quad (13.6)$$

Recognizing that $e^{-(j\omega t)}$ is sinusoidal, it becomes apparent that $e^{-((\sigma+a)t)}$ is going to determine whether this blows up or not. What we find is that if $\sigma + a$ is positive, the exponential will be to a negative power, which will cause it to go to zero as t tends to infinity. On the other hand, if $\sigma + a$ is negative or zero, the exponential will not be to a negative power, which will prevent it from tending to zero and the system will not converge. What all of this tells us is that for a causal signal, we have convergence when

Condition for Convergence

$$\text{Re}(s) > -a \quad (13.7)$$

Although we will not go through the process again for anticausal signals, we could. In doing so, we would find that the necessary condition for convergence is when

Necessary Condition for Anti-Causal Convergence

$$\text{Re}(s) < -a \quad (13.8)$$

13.4.1 Graphical Understanding of ROC

Perhaps the best way to look at the region of convergence is to view it in the s -plane. What we observe is that for a single pole, the region of convergence lies to the right of it for causal signals and to the left for anti-causal signals.

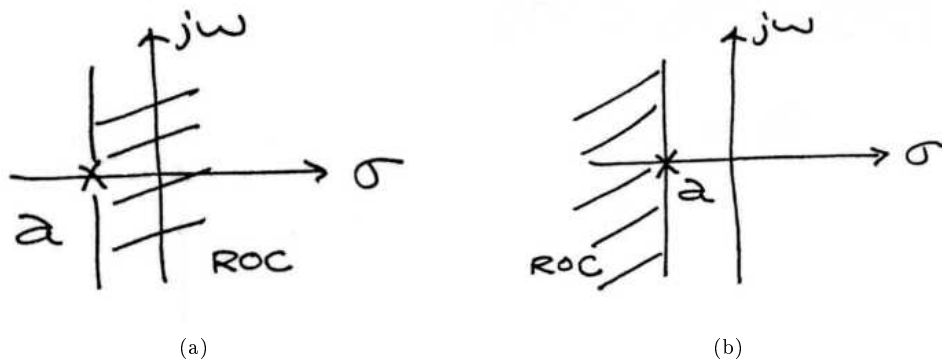


Figure 13.3: (a) The Region of Convergence for a causal signal. (b) The Region of Convergence for an anti-causal signal.

Once we have recognized this, the natural question becomes: What do we do when we have multiple poles? The simple answer is that we take the intersection of all of the regions of convergence of the respective poles.

Example 13.4

Find $H(s)$ and state the region of convergence for $h(t) = e^{-(at)}u(t) + e^{-(bt)}u(-t)$

Breaking this up into its two terms, we get transfer functions and respective regions of convergence of

$$H_1(s) = \frac{1}{s+a}, \quad \text{Re}(s) > -a \quad (13.9)$$

and

$$H_2(s) = \frac{-1}{s+b}, \quad \text{Re}(s) < -b \quad (13.10)$$

Combining these, we get a region of convergence of $-b > \text{Re}(s) > -a$. If $a > b$, we can represent this graphically. Otherwise, there will be no region of convergence.

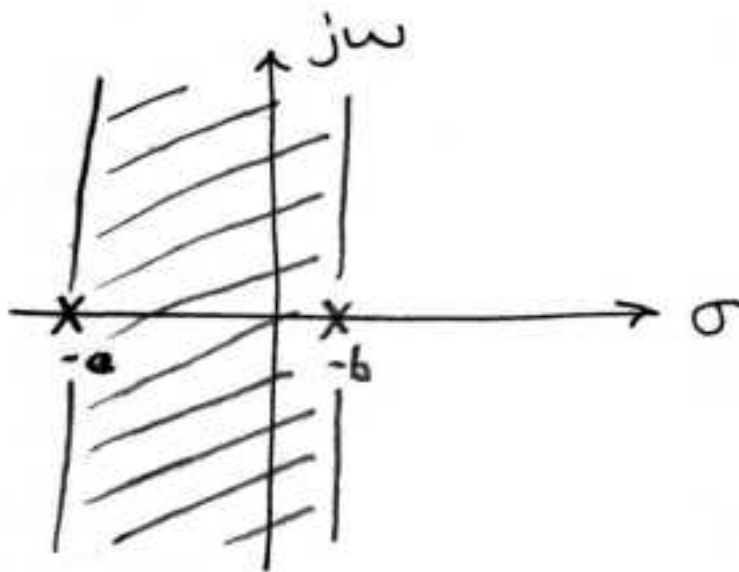


Figure 13.4: The Region of Convergence of $h(t)$ if $a > b$.

13.5 The Inverse Laplace Transform⁶

13.5.1 To Come

In The Transfer Function⁷ we shall establish that the inverse Laplace transform of a function h is

$$(\mathcal{L}^{-1}(h))(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{(c+yj)t} h((c+yj)t) dy \quad (13.11)$$

where $j \equiv \sqrt{-1}$ and the real number c is chosen so that all of the **singularities** of h lie to the left of the line of integration.

⁶This content is available online at <http://cnx.org/content/m10170/2.8/>.

⁷"Eigenvalue Problem: The Transfer Function" <http://cnx.org/content/m10490/latest/>

13.5.2 Proceeding with the Inverse Laplace Transform

With the inverse Laplace transform one may express the solution of $\mathbf{x}' = B\mathbf{x} + \mathbf{g}$, as

$$x(t) = \mathcal{L}^{-1} \left((sI - B)^{-1} \right) (\mathcal{L}\{\mathbf{g}\} + x(0)) \quad (13.12)$$

As an example, let us take the first component of $\mathcal{L}\{x\}$, namely

$$\mathcal{L}_{x_1}(s) = \frac{0.19(s^2 + 1.5s + 0.27)}{\left(s + \frac{1}{6}\right)^4 (s^3 + 1.655s^2 + 0.4078s + 0.0039)}.$$

We define:

Definition 12: poles

Also called singularities, these are the points s at which $\mathcal{L}_{x_1}(s)$ blows up.

These are clearly the roots of its denominator, namely

$$-1/100, \quad \left(-329/400 \pm \frac{\sqrt{73}}{16}\right), \quad \text{and} \quad -1/6. \quad (13.13)$$

All four being negative, it suffices to take $c = 0$ and so the integration in (13.11) proceeds up the imaginary axis. We don't suppose the reader to have already encountered integration in the complex plane but hope that this example might provide the motivation necessary for a brief overview of such. Before that however we note that MATLAB has digested the calculus we wish to develop. Referring again to fib3.m⁸ for details we note that the `ilaplace` command produces

$$x_1(t) = 211.35e^{-\frac{t}{100}} - (0.0554t^3 + 4.5464t^2 + 1.085t + 474.19)e^{-\frac{t}{6}} + e^{-\frac{(329t)}{400}} \left(262.842 \cosh\left(\frac{\sqrt{73}t}{16}\right) \right) + 262.836 \sinh\left(\frac{\sqrt{73}t}{16}\right)$$

⁸<http://www.caam.rice.edu/~caam335/cox/lectures/fib3.m>

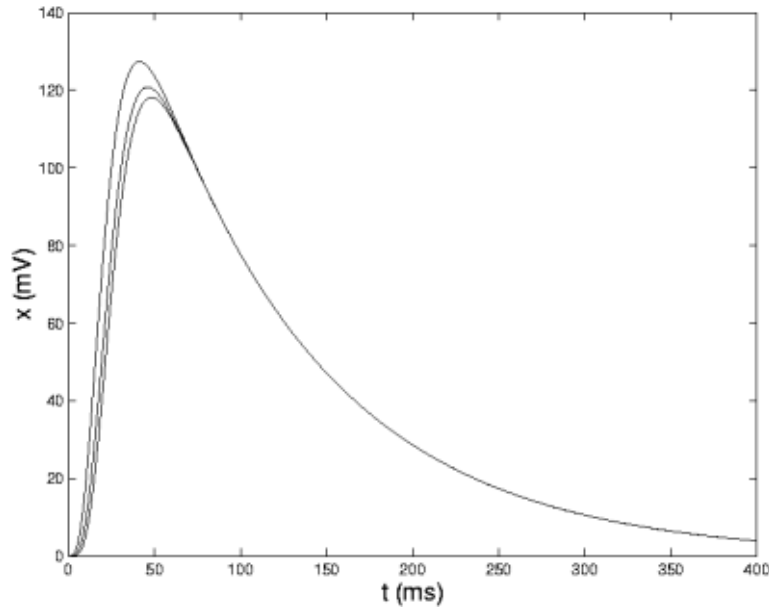


Figure 13.5: The 3 potentials associated with the RC circuit model figure⁹.

The other potentials, see the figure above, possess similar expressions. Please note that each of the poles of $\mathcal{L}\{x_1\}$ appear as exponents in x_1 and that the coefficients of the exponentials are polynomials whose degrees is determined by the **order** of the respective pole.

13.6 Poles and Zeros¹⁰

13.6.1 Introduction

It is quite difficult to qualitatively analyze the Laplace transform (Section 13.1) and Z-transform (Section 14.1), since mappings of their magnitude and phase or real part and imaginary part result in multiple mappings of 2-dimensional surfaces in 3-dimensional space. For this reason, it is very common to examine a plot of a transfer function's¹¹ poles and zeros to try to gain a qualitative idea of what a system does.

Given a continuous-time transfer function in the Laplace domain, $H(s)$, or a discrete-time one in the Z-domain, $H(z)$, a zero is any value of s or z such that the transfer function is zero, and a pole is any value of s or z such that the transfer function is infinite. To define them precisely:

Definition 13: zeros

1. The value(s) for z where the *numerator* of the transfer function equals zero
2. The complex frequencies that make the overall gain of the filter transfer function zero.

Definition 14: poles

1. The value(s) for z where the *denominator* of the transfer function equals zero
2. The complex frequencies that make the overall gain of the filter transfer function infinite.

⁹"Nerve Fibers and the Dynamic Strang Quartet", Figure 1: An RC model of a nerve fiber
http://cnx.org/content/m10168/latest/#RC_model_fig

¹⁰This content is available online at <http://cnx.org/content/m10112/2.11/>.

¹¹"Transfer Functions" <http://cnx.org/content/m0028/latest/>

13.6.2 Pole/Zero Plots

When we plot these in the appropriate s - or z -plane, we represent zeros with "o" and poles with "x". Refer to this module (Section 14.7) for a detailed looking at plotting the poles and zeros of a z -transform on the Z -plane.

Example 13.5

Find the poles and zeros for the transfer function $H(s) = \frac{s^2+6s+8}{s^2+2}$ and plot the results in the s -plane.

The first thing we recognize is that this transfer function will equal zero whenever the top, $s^2 + 6s + 8$, equals zero. To find where this equals zero, we factor this to get, $(s + 2)(s + 4)$. This yields zeros at $s = -2$ and $s = -4$. Had this function been more complicated, it might have been necessary to use the quadratic formula.

For poles, we must recognize that the transfer function will be infinite whenever the bottom part is zero. That is when $s^2 + 2$ is zero. To find this, we again look to factor the equation. This yields $(s + j\sqrt{2})(s - j\sqrt{2})$. This yields purely imaginary roots of $+j\sqrt{2}$ and $-j\sqrt{2}$.

Plotting this gives Figure 13.6 (Pole and Zero Plot)

Pole and Zero Plot

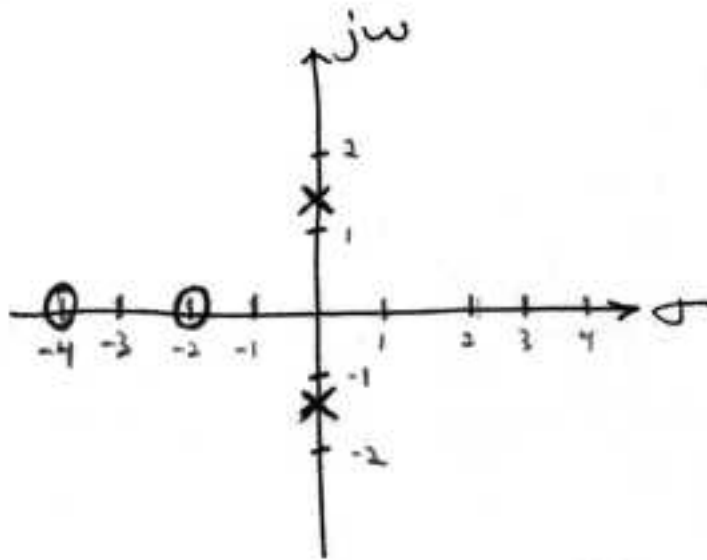


Figure 13.6: Sample pole-zero plot

Now that we have found and plotted the poles and zeros, we must ask what it is that this plot gives us. Basically what we can gather from this is that the magnitude of the transfer function will be larger when it is closer to the poles and smaller when it is closer to the zeros. This provides us with a qualitative understanding of what the system does at various frequencies and is crucial to the discussion of stability (Section 3.4).

13.6.3 Repeated Poles and Zeros

It is possible to have more than one pole or zero at any given point. For instance, the discrete-time transfer function $H(z) = z^2$ will have two zeros at the origin and the continuous-time function $H(s) = \frac{1}{s^{25}}$ will have

25 poles at the origin.

13.6.4 Pole-Zero Cancellation

An easy mistake to make with regards to poles and zeros is to think that a function like $\frac{(s+3)(s-1)}{s-1}$ is the same as $s + 3$. In theory they are equivalent, as the pole and zero at $s = 1$ cancel each other out in what is known as **pole-zero cancellation**. However, think about what may happen if this were a transfer function of a system that was created with physical circuits. In this case, it is very unlikely that the pole and zero would remain in exactly the same place. A minor temperature change, for instance, could cause one of them to move just slightly. If this were to occur a tremendous amount of volatility is created in that area, since there is a change from infinity at the pole to zero at the zero in a very small range of signals. This is generally a very bad way to try to eliminate a pole. A much better way is to use **control theory** to move the pole to a better place.

Chapter 14

Z-Transform and Digital Filtering

14.1 The Z Transform: Definition¹

14.1.1 Basic Definition of the Z-Transform

The **z-transform** of a sequence is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} (x[n] z^{-n}) \quad (14.1)$$

Sometimes this equation is referred to as the **bilateral z-transform**. At times the z-transform is defined as

$$X(z) = \sum_{n=0}^{\infty} (x[n] z^{-n}) \quad (14.2)$$

which is known as the **unilateral z-transform**.

There is a close relationship between the z-transform and the **Fourier transform** of a discrete time signal, which is defined as

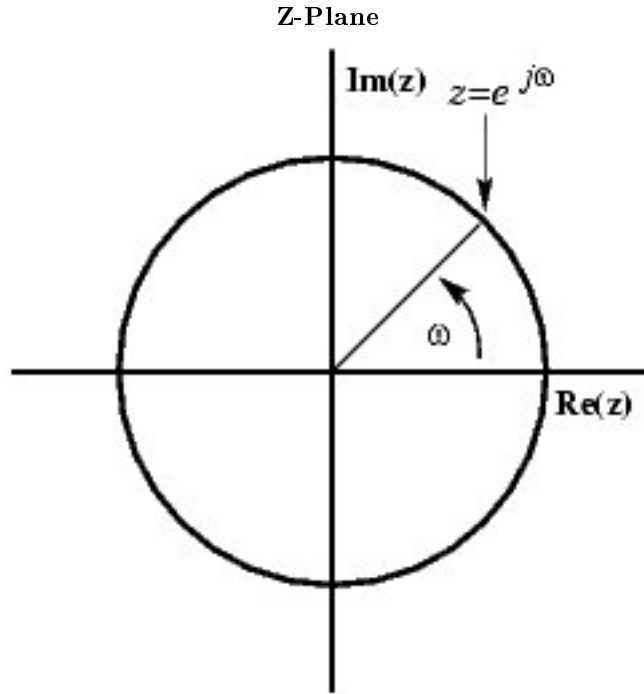
$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} (x[n] e^{-j\omega n}) \quad (14.3)$$

Notice that that when the z^{-n} is replaced with $e^{-j\omega n}$ the z-transform reduces to the Fourier Transform. When the Fourier Transform exists, $z = e^{j\omega}$, which is to have the magnitude of z equal to unity.

14.1.2 The Complex Plane

In order to get further insight into the relationship between the Fourier Transform and the Z-Transform it is useful to look at the complex plane or **z-plane**. Take a look at the complex plane:

¹This content is available online at <<http://cnx.org/content/m10549/2.9/>>.



The Z-plane is a complex plane with an imaginary and real axis referring to the complex-valued variable z . The position on the complex plane is given by $re^{j\omega}$, and the angle from the positive, real axis around the plane is denoted by ω . $X(z)$ is defined everywhere on this plane. $X(e^{j\omega})$ on the other hand is defined only where $|z| = 1$, which is referred to as the unit circle. So for example, $\omega = 0$ at $z = 1$ and $\omega = \pi$ at $z = -1$. This is useful because, by representing the Fourier transform as the z-transform on the unit circle, the periodicity of Fourier transform is easily seen.

14.1.3 Region of Convergence

The region of convergence, known as the **ROC**, is important to understand because it defines the region where the z-transform exists. The ROC for a given $x[n]$, is defined as the range of z for which the z-transform converges. Since the z-transform is a **power series**, it converges when $x[n]z^{-n}$ is absolutely summable. Stated differently,

$$\sum_{n=-\infty}^{\infty} (|x[n]z^{-n}|) < \infty \quad (14.4)$$

must be satisfied for convergence. This is best illustrated by looking at the different ROC's of the z-transforms of $\alpha^n u[n]$ and $\alpha^n u[n-1]$.

Example 14.1

For

$$x[n] = \alpha^n u[n] \quad (14.5)$$

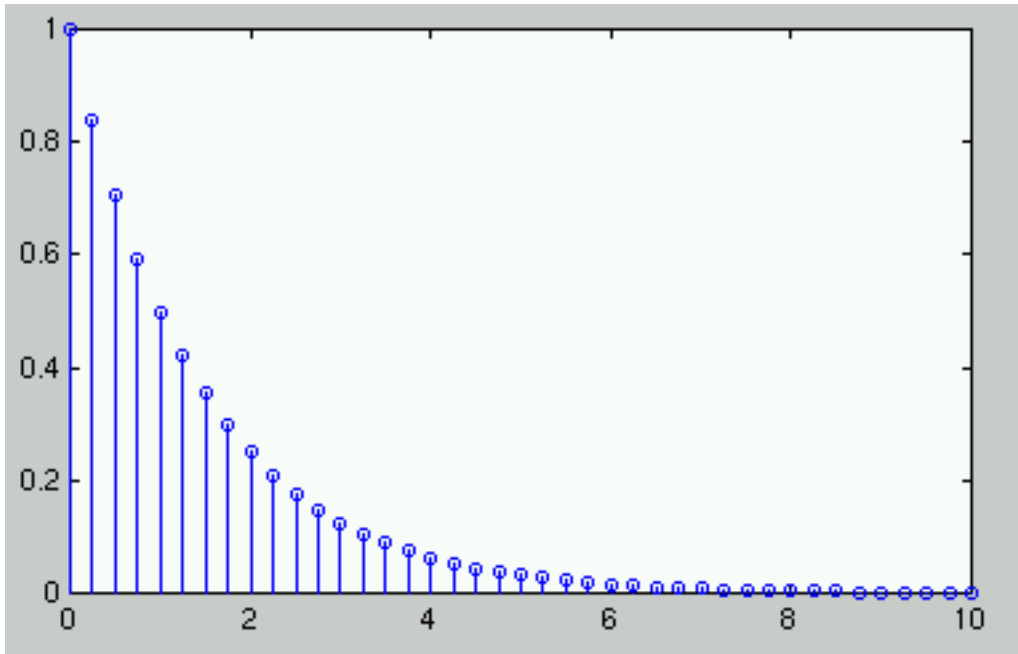


Figure 14.2: $x[n] = \alpha^n u[n]$ where $\alpha = 0.5$.

$$\begin{aligned}
 X(z) &= \sum_{n=-\infty}^{\infty} (x[n] z^{-n}) \\
 &= \sum_{n=-\infty}^{\infty} (\alpha^n u[n] z^{-n}) \\
 &= \sum_{n=0}^{\infty} (\alpha^n z^{-n}) \\
 &= \sum_{n=0}^{\infty} ((\alpha z^{-1})^n)
 \end{aligned} \tag{14.6}$$

This sequence is an example of a right-sided exponential sequence because it is nonzero for $n \geq 0$. It only converges when $|\alpha z^{-1}| < 1$. When it converges,

$$\begin{aligned}
 X(z) &= \frac{1}{1 - \alpha z^{-1}} \\
 &= \frac{z}{z - \alpha}
 \end{aligned} \tag{14.7}$$

If $|\alpha z^{-1}| \geq 1$, then the series, $\sum_{n=0}^{\infty} ((\alpha z^{-1})^n)$ does not converge. Thus the ROC is the range of values where

$$|\alpha z^{-1}| < 1 \tag{14.8}$$

or, equivalently,

$$|z| > |\alpha| \tag{14.9}$$

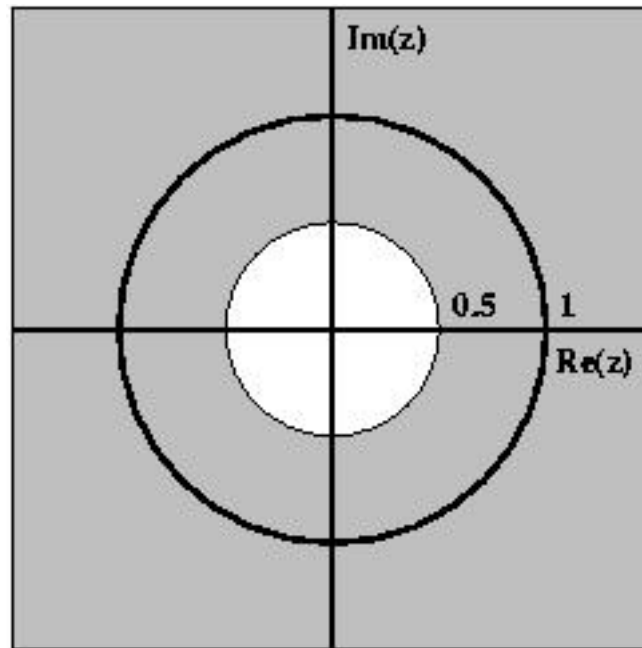


Figure 14.3: ROC for $x[n] = \alpha^n u[n]$ where $\alpha = 0.5$

Example 14.2

For

$$x[n] = (-\alpha^n) u[-n-1] \quad (14.10)$$

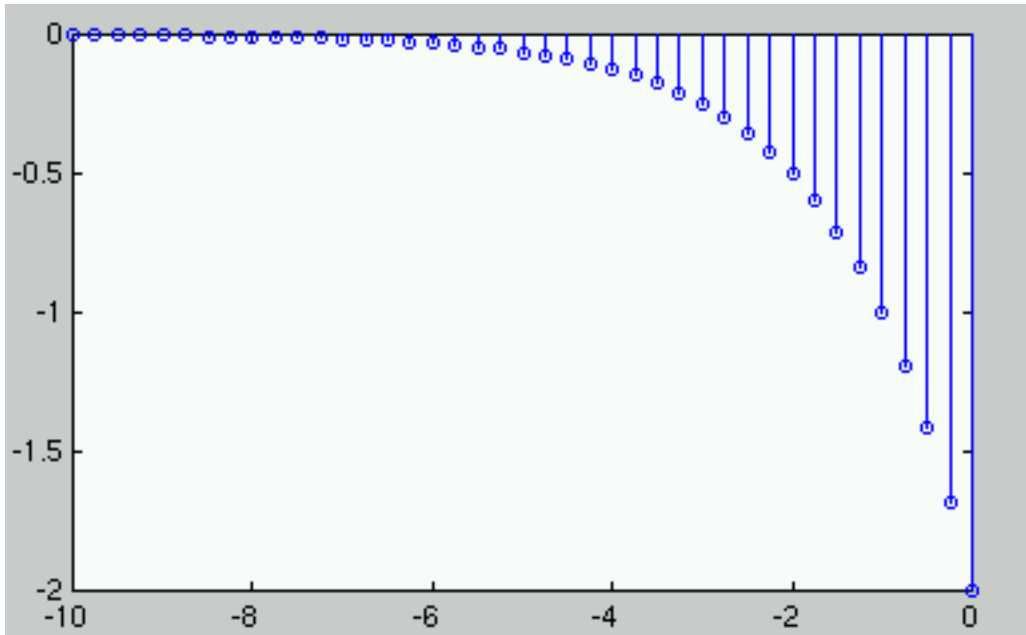


Figure 14.4: $x[n] = -(\alpha^n)u[-n-1]$ where $\alpha = 0.5$.

$$\begin{aligned}
 X(z) &= \sum_{n=-\infty}^{\infty} (x[n] z^{-n}) \\
 &= \sum_{n=-\infty}^{\infty} ((-\alpha^n) u[-n-1] z^{-n}) \\
 &= -\left(\sum_{n=-\infty}^{-1} (\alpha^n z^{-n})\right) \\
 &= -\left(\sum_{n=-\infty}^{-1} ((\alpha^{-1}z)^{-n})\right) \\
 &= -\left(\sum_{n=1}^{\infty} ((\alpha^{-1}z)^n)\right) \\
 &= 1 - \sum_{n=0}^{\infty} ((\alpha^{-1}z)^n)
 \end{aligned} \tag{14.11}$$

The ROC in this case is the range of values where

$$|\alpha^{-1}z| < 1 \tag{14.12}$$

or, equivalently,

$$|z| < |\alpha| \tag{14.13}$$

If the ROC is satisfied, then

$$\begin{aligned}
 X(z) &= 1 - \frac{1}{1 - \alpha^{-1}z} \\
 &= \frac{z}{z - \alpha}
 \end{aligned} \tag{14.14}$$

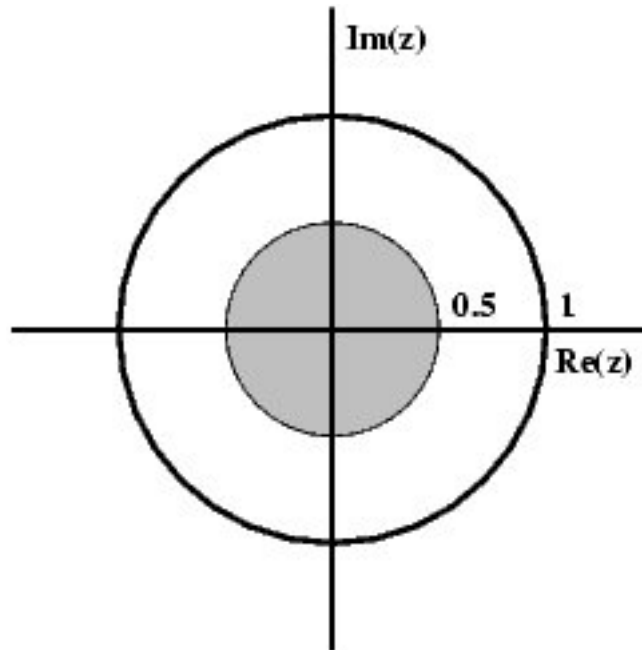


Figure 14.5: ROC for $x[n] = -(\alpha^n)u[-n-1]$

14.2 Table of Common z-Transforms²

The table below provides a number of unilateral and bilateral **z-transforms** (Section 14.1). The table also specifies the region of convergence (Section 14.3).

NOTE: The notation for z found in the table below may differ from that found in other tables. For example, the basic z-transform of $u[n]$ can be written as either of the following two expressions, which are equivalent:

$$\frac{z}{z-1} = \frac{1}{1-z^{-1}} \quad (14.15)$$

²This content is available online at <<http://cnx.org/content/m10119/2.13/>>.

Signal	Z-Transform	ROC
$\delta[n - k]$	z^{-k}	All z
$u[n]$	$\frac{z}{z-1}$	$ z > 1$
$-(u[-n - 1])$	$\frac{z}{z-1}$	$ z < 1$
$nu[n]$	$\frac{z}{(z-1)^2}$	$ z > 1$
$n^2u[n]$	$\frac{z(z+1)}{(z-1)^3}$	$ z > 1$
$n^3u[n]$	$\frac{z(z^2+4z+1)}{(z-1)^4}$	$ z > 1$
$(-\alpha^n)u[-n - 1]$	$\frac{z}{z-\alpha}$	$ z < \alpha $
$\alpha^n u[n]$	$\frac{z}{z-\alpha}$	$ z > \alpha $
$n\alpha^n u[n]$	$\frac{\alpha z}{(z-\alpha)^2}$	$ z > \alpha $
$n^2\alpha^n u[n]$	$\frac{\alpha z(z+\alpha)}{(z-\alpha)^3}$	$ z > \alpha $
$\frac{\prod_{k=1}^m (n-k+1)}{\alpha^m m!} \alpha^n u[n]$	$\frac{z}{(z-\alpha)^{m+1}}$	
$\gamma^n \cos(\alpha n) u[n]$	$\frac{z(z-\gamma \cos(\alpha))}{z^2 - (2\gamma \cos(\alpha))z + \gamma^2}$	$ z > \gamma $
$\gamma^n \sin(\alpha n) u[n]$	$\frac{z\gamma \sin(\alpha)}{z^2 - (2\gamma \cos(\alpha))z + \gamma^2}$	$ z > \gamma $

14.3 Region of Convergence for the Z-transform³

14.3.1 The Region of Convergence

The region of convergence, known as the **ROC**, is important to understand because it defines the region where the z-transform (Section 14.1) exists. The **z-transform** of a sequence is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} (x[n] z^{-n}) \quad (14.16)$$

The ROC for a given $x[n]$, is defined as the range of z for which the z-transform converges. Since the z-transform is a **power series**, it converges when $x[n] z^{-n}$ is absolutely summable. Stated differently,

$$\sum_{n=-\infty}^{\infty} (|x[n] z^{-n}|) < \infty \quad (14.17)$$

must be satisfied for convergence.

14.3.2 Properties of the Region of Convergence

The Region of Convergence has a number of properties that are dependent on the characteristics of the signal, $x[n]$.

- *The ROC cannot contain any poles.* By definition a pole is a where $X(z)$ is infinite. Since $X(z)$ must be finite for all z for convergence, there cannot be a pole in the ROC.
- *If $x[n]$ is a finite-duration sequence, then the ROC is the entire z-plane, except possibly $z = 0$ or $|z| = \infty$.* A **finite-duration sequence** is a sequence that is nonzero in a finite interval $n_1 \leq n \leq n_2$. As long as each value of $x[n]$ is finite then the sequence will be absolutely summable. When $n_2 > 0$

³This content is available online at <<http://cnx.org/content/m10622/2.5/>>.

there will be a z^{-1} term and thus the ROC will not include $z = 0$. When $n_1 < 0$ then the sum will be infinite and thus the ROC will not include $|z| = \infty$. On the other hand, when $n_2 \leq 0$ then the ROC will include $z = 0$, and when $n_1 \geq 0$ the ROC will include $|z| = \infty$. With these constraints, the only signal, then, whose ROC is the entire z -plane is $x[n] = c\delta[n]$.

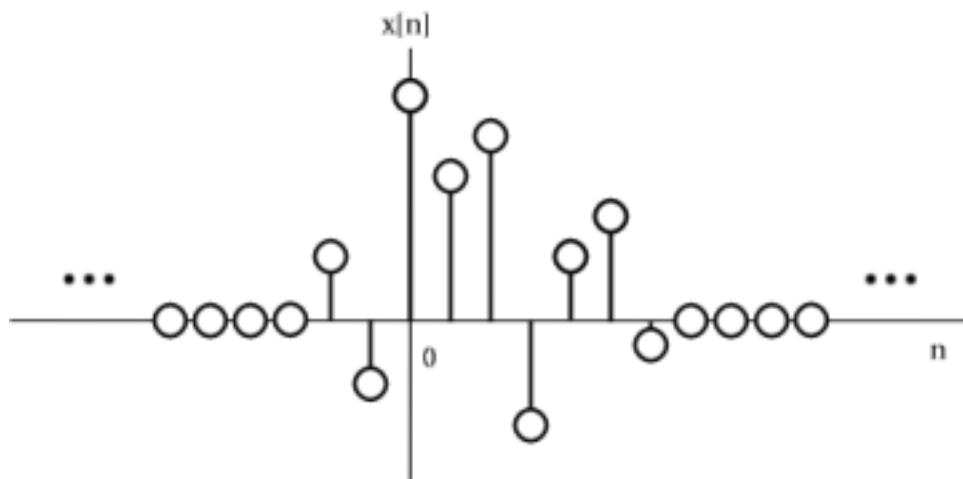


Figure 14.6: An example of a finite duration sequence.

The next properties apply to infinite duration sequences. As noted above, the z -transform converges when $|X(z)| < \infty$. So we can write

$$|X(z)| = \left| \sum_{n=-\infty}^{\infty} (x[n] z^{-n}) \right| \leq \sum_{n=-\infty}^{\infty} (|x[n] z^{-n}|) = \sum_{n=-\infty}^{\infty} (|x[n]| |z|^{-n}) \quad (14.18)$$

We can then split the infinite sum into positive-time and negative-time portions. So

$$|X(z)| \leq N(z) + P(z) \quad (14.19)$$

where

$$N(z) = \sum_{n=-\infty}^{-1} (|x[n]| |z|^{-n}) \quad (14.20)$$

and

$$P(z) = \sum_{n=0}^{\infty} (|x[n]| |z|^{-n}) \quad (14.21)$$

In order for $|X(z)|$ to be finite, $|x[n]|$ must be bounded. Let us then set

$$|x(n)| \leq C_1 r_1^n \quad (14.22)$$

for

$$n < 0$$

and

$$|x(n)| \leq C_2 r_2^n \quad (14.23)$$

for

$$n \geq 0$$

From this some further properties can be derived:

- If $x[n]$ is a right-sided sequence, then the ROC extends outward from the outermost pole in $X(z)$. A **right-sided sequence** is a sequence where $x[n] = 0$ for $n < n_1 < \infty$. Looking at the positive-time portion from the above derivation, it follows that

$$P(z) \leq C_2 \sum_{n=0}^{\infty} (r_2^n (|z|)^{-n}) = C_2 \sum_{n=0}^{\infty} \left(\left(\frac{r_2}{|z|} \right)^n \right) \quad (14.24)$$

Thus in order for this sum to converge, $|z| > r_2$, and therefore the ROC of a right-sided sequence is of the form $|z| > r_2$.

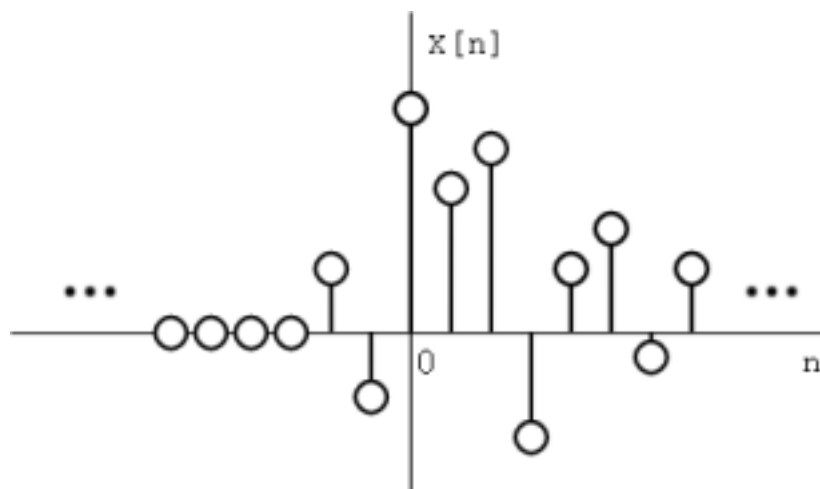


Figure 14.7: A right-sided sequence.

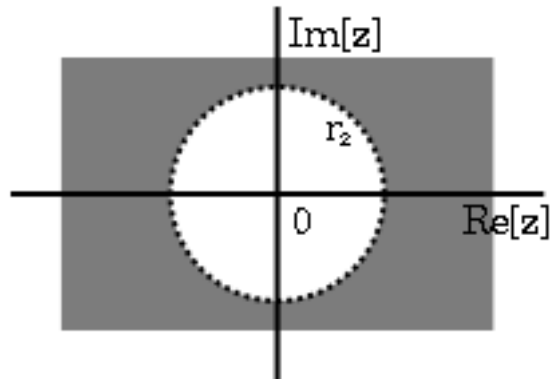


Figure 14.8: The ROC of a right-sided sequence.

- If $x[n]$ is a left-sided sequence, then the ROC extends inward from the innermost pole in $X(z)$. A **right-sided sequence** is a sequence where $x[n] = 0$ for $n > n_2 > -\infty$. Looking at the negative-time portion from the above derivation, it follows that

$$N(z) \leq C_1 \sum_{n=-\infty}^{-1} (r_1^n (|z|)^{-n}) = C_1 \sum_{n=-\infty}^{-1} \left(\left(\frac{r_1}{|z|} \right)^n \right) = C_1 \sum_{k=1}^{\infty} \left(\left(\frac{|z|}{r_1} \right)^k \right) \quad (14.25)$$

Thus in order for this sum to converge, $|z| < r_1$, and therefore the ROC of a left-sided sequence is of the form $|z| < r_1$.

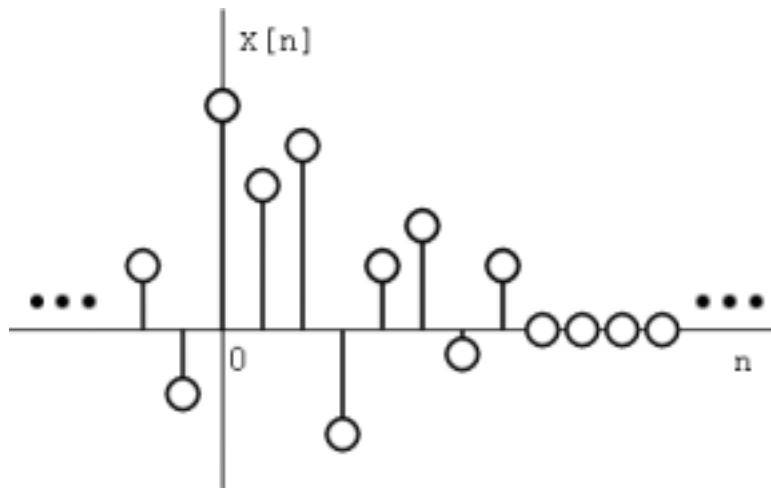


Figure 14.9: A left-sided sequence.

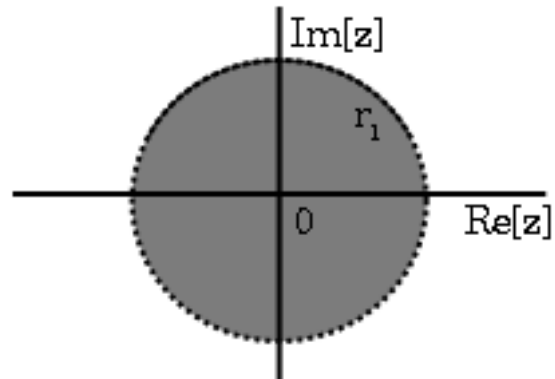


Figure 14.10: The ROC of a left-sided sequence.

- If $x[n]$ is a two-sided sequence, the ROC will be a ring in the z -plane that is bounded on the interior and exterior by a pole. A **two-sided sequence** is an sequence with infinite duration in the positive and negative directions. From the derivation of the above two properties, it follows that if $r_2 < |z| < r_2$ converges, then both the positive-time and negative-time portions converge and thus $X(z)$ converges as well. Therefore the ROC of a two-sided sequence is of the form $r_2 < |z| < r_2$.

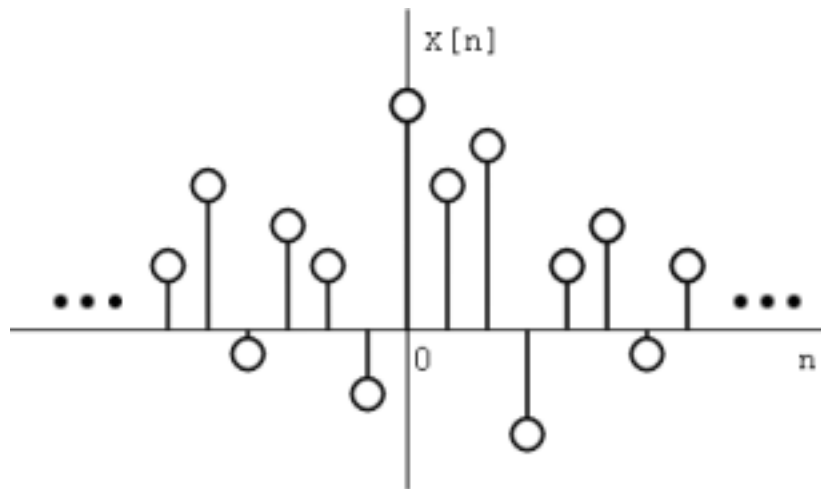


Figure 14.11: A two-sided sequence.

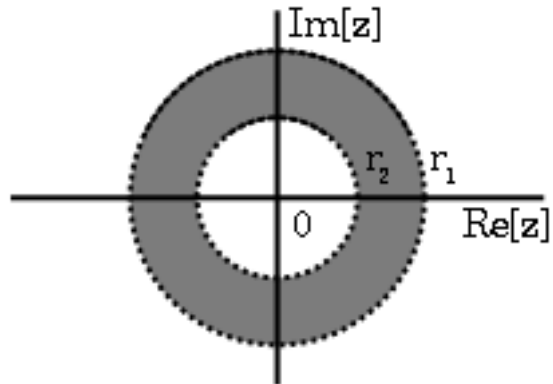


Figure 14.12: The ROC of a two-sided sequence.

14.3.3 Examples

To gain further insight it is good to look at a couple of examples.

Example 14.3

Lets take

$$x_1[n] = \left(\frac{1}{2}\right)^n u[n] + \left(\frac{1}{4}\right)^n u[n] \quad (14.26)$$

The z-transform of $\left(\frac{1}{2}\right)^n u[n]$ is $\frac{z}{z-\frac{1}{2}}$ with an ROC at $|z| > \frac{1}{2}$.

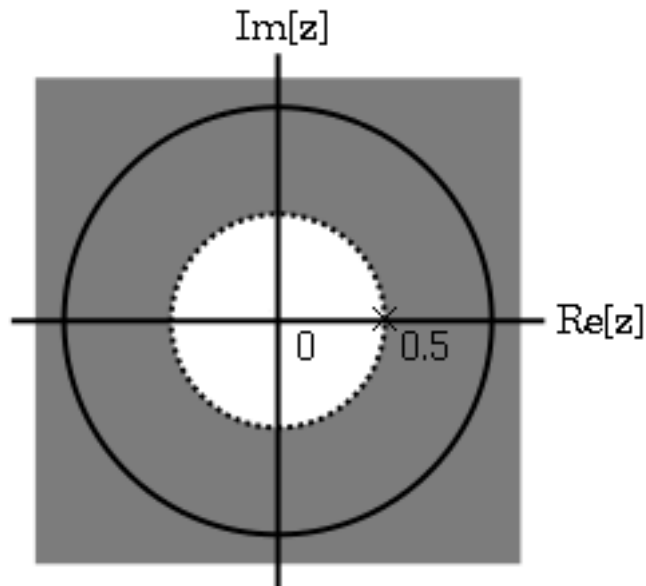


Figure 14.13: The ROC of $\left(\frac{1}{2}\right)^n u[n]$

The z-transform of $(\frac{-1}{4})^n u[n]$ is $\frac{z}{z+\frac{1}{4}}$ with an ROC at $|z| > \frac{-1}{4}$.

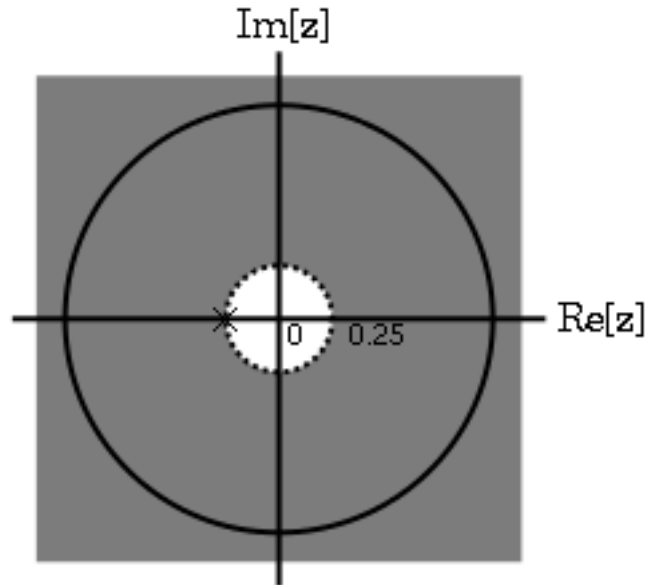


Figure 14.14: The ROC of $(\frac{-1}{4})^n u[n]$

Due to linearity,

$$\begin{aligned} X_1[z] &= \frac{z}{z-\frac{1}{2}} + \frac{z}{z+\frac{1}{4}} \\ &= \frac{2z(z-\frac{1}{8})}{(z-\frac{1}{2})(z+\frac{1}{4})} \end{aligned} \quad (14.27)$$

By observation it is clear that there are two zeros, at 0 and $\frac{1}{8}$, and two poles, at $\frac{1}{2}$, and $\frac{-1}{4}$. Following the above properties, the ROC is $|z| > \frac{1}{2}$.

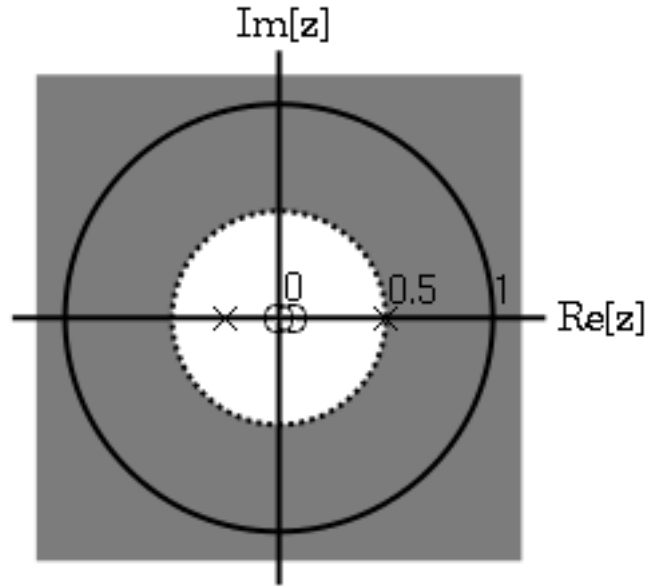


Figure 14.15: The ROC of $x_1[n] = \left(\frac{1}{2}\right)^n u[n] + \left(\frac{-1}{4}\right)^n u[n]$

Example 14.4

Now take

$$x_2[n] = \left(\frac{-1}{4}\right)^n u[n] - \left(\frac{1}{2}\right)^n u[-n-1] \quad (14.28)$$

The z-transform and ROC of $\left(\frac{-1}{4}\right)^n u[n]$ was shown in the example above (Example 14.3). The z-transform of $\left(-\left(\frac{1}{2}\right)^n\right) u[-n-1]$ is $\frac{z}{z-\frac{1}{2}}$ with an ROC at $|z| > \frac{1}{2}$.

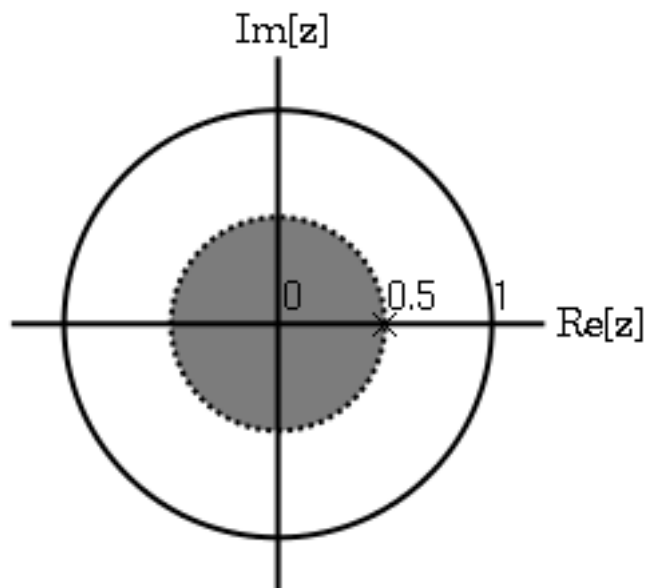


Figure 14.16: The ROC of $(-\left(\frac{1}{2}\right)^n)u[-n-1]$

Once again, by linearity,

$$\begin{aligned} X_2[z] &= \frac{z}{z+\frac{1}{4}} + \frac{z}{z-\frac{1}{2}} \\ &= \frac{z(2z-\frac{1}{8})}{(z+\frac{1}{4})(z-\frac{1}{2})} \end{aligned} \quad (14.29)$$

By observation it is again clear that there are two zeros, at 0 and $\frac{1}{16}$, and two poles, at $\frac{1}{2}$, and $-\frac{1}{4}$. in this case though, the ROC is $|z| < \frac{1}{2}$.

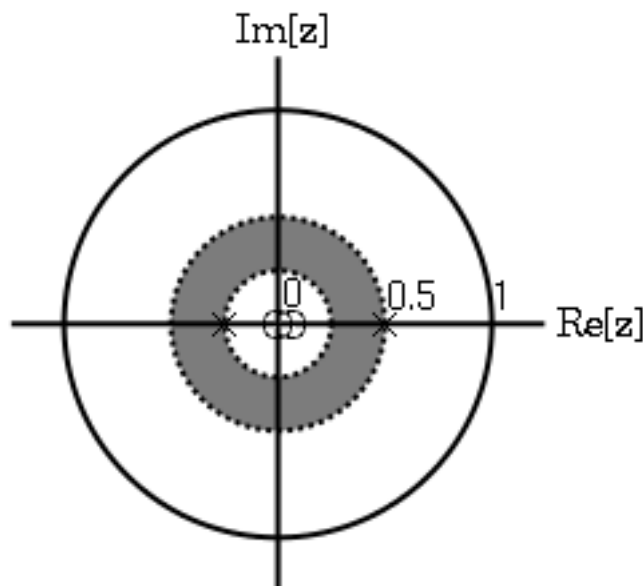


Figure 14.17: The ROC of $x_2[n] = \left(\frac{-1}{4}\right)^n u[n] - \left(\frac{1}{2}\right)^n u[-n-1]$.

14.4 Inverse Z-Transform⁴

When using the z-transform (Section 14.1)

$$X(z) = \sum_{n=-\infty}^{\infty} (x[n] z^{-n}) \quad (14.30)$$

it is often useful to be able to find $x[n]$ given $X(z)$. There are at least 4 different methods to do this:

1. Inspection (Section 14.4.1: Inspection Method)
2. Partial-Fraction Expansion (Section 14.4.2: Partial-Fraction Expansion Method)
3. Power Series Expansion (Section 14.4.3: Power Series Expansion Method)
4. Contour Integration (Section 14.4.4: Contour Integration Method)

14.4.1 Inspection Method

This "method" is to basically become familiar with the z-transform pair tables (Section 14.2) and then "reverse engineer".

Example 14.5

When given

$$X(z) = \frac{z}{z - \alpha}$$

with an ROC (Section 14.3) of

$$|z| > \alpha$$

⁴This content is available online at <<http://cnx.org/content/m10651/2.4/>>.

we could determine "by inspection" that

$$x[n] = \alpha^n u[n]$$

14.4.2 Partial-Fraction Expansion Method

When dealing with **linear time-invariant systems** the z-transform often in the form

$$\begin{aligned} X(z) &= \frac{B(z)}{A(z)} \\ &= \frac{\sum_{k=0}^M (b_k z^{-k})}{\sum_{k=0}^N (a_k z^{-k})} \end{aligned} \quad (14.31)$$

This can also expressed as

$$X(z) = \frac{a_0 \prod_{k=1}^M (1 - c_k z^{-1})}{b_0 \prod_{k=1}^N (1 - d_k z^{-1})} \quad (14.32)$$

where c_k represents the nonzero zeros of $X(z)$ and d_k represents the nonzero poles.

If $M < N$ then $X(z)$ can be represented as

$$X(z) = \sum_{k=1}^N \left(\frac{A_k}{1 - d_k z^{-1}} \right) \quad (14.33)$$

This form allows for easy inversions of each term of the sum using the inspection method (Section 14.4.1: Inspection Method) and the transform table (Section 14.2). Thus if the numerator is a polynomial then it is necessary to use partial-fraction expansion⁵ to put $X(z)$ in the above form. If $M \geq N$ then $X(z)$ can be expressed as

$$X(z) = \sum_{r=0}^{M-N} (B_r z^{-r}) + \frac{\sum_{k=0}^{N-1} (b'_k z^{-k})}{\sum_{k=0}^N (a_k z^{-k})} \quad (14.34)$$

Example 14.6

Find the inverse z-transform of

$$X(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 + (-3z^{-1}) + 2z^{-2}}$$

where the ROC is $|z| > 2$. In this case $M = N = 2$, so we have to use long division to get

$$X(z) = \frac{1}{2} + \frac{\frac{1}{2} + \frac{7}{2}z^{-1}}{1 + (-3z^{-1}) + 2z^{-2}}$$

Next factor the denominator.

$$X(z) = 2 + \frac{(-1) + 5z^{-1}}{(1 - 2z^{-1})(1 - z^{-1})}$$

Now do partial-fraction expansion.

$$X(z) = \frac{1}{2} + \frac{A_1}{1 - 2z^{-1}} + \frac{A_2}{1 - z^{-1}} = \frac{1}{2} + \frac{\frac{9}{2}}{1 - 2z^{-1}} + \frac{-4}{1 - z^{-1}}$$

Now each term can be inverted using the inspection method and the z-transform table. Thus, since the ROC is $|z| > 2$,

$$x[n] = \frac{1}{2}\delta[n] + \frac{9}{2}2^n u[n] + (-4u[n])$$

⁵"Partial Fraction Expansion" <<http://cnx.org/content/m2111/latest/>>

14.4.3 Power Series Expansion Method

When the z-transform is defined as a power series in the form

$$X(z) = \sum_{n=-\infty}^{\infty} (x[n] z^{-n}) \quad (14.35)$$

then each term of the sequence $x[n]$ can be determined by looking at the coefficients of the respective power of z^{-n} .

Example 14.7

Now look at the z-transform of a **finite-length sequence**.

$$\begin{aligned} X(z) &= z^2 (1 + 2z^{-1}) (1 - \frac{1}{2}z^{-1}) (1 + z^{-1}) \\ &= z^2 + \frac{5}{2}z + \frac{1}{2} + (-z^{-1}) \end{aligned} \quad (14.36)$$

In this case, since there were no poles, we multiplied the factors of $X(z)$. Now, by inspection, it is clear that

$$x[n] = \delta[n+2] + \frac{5}{2}\delta[n+1] + \frac{1}{2}\delta[n] + (-\delta[n-1])$$

One of the advantages of the power series expansion method is that many functions encountered in engineering problems have their power series' tabulated. Thus functions such as log, sin, exponent, sinh, etc, can be easily inverted.

Example 14.8

Suppose

$$X(z) = \log_n (1 + \alpha z^{-1})$$

Noting that

$$\log_n (1 + x) = \sum_{n=1}^{\infty} \left(\frac{-1^{n+1} x^n}{n} \right)$$

Then

$$X(z) = \sum_{n=1}^{\infty} \left(\frac{-1^{n+1} \alpha^n z^{-n}}{n} \right)$$

Therefore

$$X(z) = \begin{cases} \frac{-1^{n+1} \alpha^n}{n} & \text{if } n \geq 1 \\ 0 & \text{if } n \leq 0 \end{cases}$$

14.4.4 Contour Integration Method

Without going in to much detail

$$x[n] = \frac{1}{2\pi j} \oint_r X(z) z^{n-1} dz \quad (14.37)$$

where r is a counter-clockwise contour in the ROC of $X(z)$ encircling the origin of the z-plane. To further expand on this method of finding the inverse requires the knowledge of complex variable theory and thus will not be addressed in this module.

14.5 Rational Functions⁶

14.5.1 Introduction

When dealing with operations on polynomials, the term **rational function** is a simple way to describe a particular relationship between two polynomials.

Definition 15: rational function

For any two polynomials, A and B, their quotient is called a rational function.

Example

Below is a simple example of a basic rational function, $f(x)$. Note that the numerator and denominator can be polynomials of any order, but the rational function is undefined when the denominator equals zero.

$$f(x) = \frac{x^2 - 4}{2x^2 + x - 3} \quad (14.38)$$

If you have begun to study the Z-transform (Section 14.1), you should have noticed by now they are all rational functions. Below we will look at some of the properties of rational functions and how they can be used to reveal important characteristics about a z-transform, and thus a signal or LTI system.

14.5.2 Properties of Rational Functions

In order to see what makes rational functions special, let us look at some of their basic properties and characteristics. If you are familiar with rational functions and basic algebraic properties, skip to the next section (Section 14.5.3: Rational Functions and the Z-Transform) to see how rational functions are useful when dealing with the z-transform.

14.5.2.1 Roots

To understand many of the following characteristics of a rational function, one must begin by finding the roots of the rational function. In order to do this, let us factor both of the polynomials so that the roots can be easily determined. Like all polynomials, the roots will provide us with information on many key properties. The function below shows the results of factoring the above rational function, (14.38).

$$f(x) = \frac{(x + 2)(x - 2)}{(2x + 3)(x - 1)} \quad (14.39)$$

Thus, the roots of the rational function are as follows:

Roots of the numerator are: $\{-2, 2\}$

Roots of the denominator are: $\{-3, 1\}$

NOTE: In order to understand rational functions, it is essential to know and understand the roots that make up the rational function.

14.5.2.2 Discontinuities

Because we are dealing with division of two polynomials, we must be aware of the values of the variable that will cause the denominator of our fraction to be zero. When this happens, the rational function becomes undefined, *i.e.* we have a discontinuity in the function. Because we have already solved for our roots, it is very easy to see when this occurs. When the variable in the denominator equals any of the roots of the denominator, the function becomes undefined.

⁶This content is available online at <<http://cnx.org/content/m10593/2.7/>>.

Example 14.9

Continuing to look at our rational function above, (14.38), we can see that the function will have discontinuities at the following points: $x = \{-3, 1\}$

In respect to the Cartesian plane, we say that the discontinuities are the values along the x-axis where the function is undefined. These discontinuities often appear as **vertical asymptotes** on the graph to represent the values where the function is undefined.

14.5.2.3 Domain

Using the roots that we found above, the **domain** of the rational function can be easily defined.

Definition 16: domain

The group, or set, of values that are defined by a given function.

Example

Using the rational function above, (14.38), the domain can be defined as any real number x where x does not equal 1 or negative 3. Written out mathematically, we get the following:

$$\{x \in \mathbb{R} \mid x \neq -3 \text{ and } x \neq 1\} \quad (14.40)$$

14.5.2.4 Intercepts

The **x-intercept** is defined as the point(s) where $f(x)$, i.e. the output of the rational functions, equals zero. Because we have already found the roots of the equation this process is very simple. From algebra, we know that the output will be zero whenever the numerator of the rational function is equal to zero. Therefore, the function will have an x-intercept wherever x equals one of the roots of the numerator.

The **y-intercept** occurs whenever x equals zero. This can be found by setting all the values of x equal to zero and solving the rational function.

14.5.3 Rational Functions and the Z-Transform

As we have stated above, all z-transforms can be written as rational functions, which have become the most common way of representing the z-transform. Because of this, we can use the properties above, especially those of the roots, in order to reveal certain characteristics about the signal or LTI system described by the z-transform.

Below is the general form of the z-transform written as a rational function:

$$X(z) = \frac{b_0 + b_1z^{-1} + \dots + b_Mz^{-M}}{a_0 + a_1z^{-1} + \dots + a_Nz^{-N}} \quad (14.41)$$

If you have already looked at the module about Understanding Pole/Zero Plots and the Z-transform (Section 14.7), you should see how the roots of the rational function play an important role in understanding the z-transform. The equation above, (14.41), can be expressed in factored form just as was done for the simple rational function above, see (14.39). Thus, we can easily find the roots of the numerator and denominator of the z-transform. The following two relationships become apparent:

Relationship of Roots to Poles and Zeros

- The roots of the numerator in the rational function will be the **zeros** of the z-transform
- The roots of the denominator in the rational function will be the **poles** of the z-transform

14.5.4 Conclusion

Once we have used our knowledge of rational functions to find its roots, we can manipulate a z-transform in a number of useful ways. We can apply this knowledge to representing an LTI system graphically through a Pole/Zero Plot (Section 14.7), or to analyze and design a digital filter through Filter Design from the Z-Transform (Section 14.8).

14.6 Difference Equation⁷

14.6.1 Introduction

One of the most important concepts of DSP is to be able to properly represent the input/output relationship to a given LTI system. A linear constant-coefficient **difference equation** (LCCDE) serves as a way to express just this relationship in a discrete-time system. Writing the sequence of inputs and outputs, which represent the characteristics of the LTI system, as a difference equation help in understanding and manipulating a system.

Definition 17: difference equation

An equation that shows the relationship between consecutive values of a sequence and the differences among them. They are often rearranged as a recursive formula so that a systems output can be computed from the input signal and past outputs.

Example

$$y[n] + 7y[n - 1] + 2y[n - 2] = x[n] - 4x[n - 1] \quad (14.42)$$

14.6.2 General Formulas from the Difference Equation

As stated briefly in the definition above, a difference equation is a very useful tool in describing and calculating the output of the system described by the formula for a given sample n . The key property of the difference equation is its ability to help easily find the transform, $H(z)$, of a system. In the following two subsections, we will look at the general form of the difference equation and the general conversion to a z-transform directly from the difference equation.

14.6.2.1 Difference Equation

The general form of a linear, constant-coefficient difference equation (LCCDE), is shown below:

$$\sum_{k=0}^N (a_k y[n - k]) = \sum_{k=0}^M (b_k x[n - k]) \quad (14.43)$$

We can also write the general form to easily express a recursive output, which looks like this:

$$y[n] = - \left(\sum_{k=1}^N (a_k y[n - k]) \right) + \sum_{k=0}^M (b_k x[n - k]) \quad (14.44)$$

From this equation, note that $y[n - k]$ represents the outputs and $x[n - k]$ represents the inputs. The value of N represents the **order** of the difference equation and corresponds to the memory of the system being represented. Because this equation relies on past values of the output, in order to compute a numerical solution, certain past outputs, referred to as the **initial conditions**, must be known.

⁷This content is available online at <<http://cnx.org/content/m10595/2.5/>>.

14.6.2.2 Conversion to Z-Transform

Using the above formula, (14.43), we can easily generalize the **transfer function**, $H(z)$, for any difference equation. Below are the steps taken to convert any difference equation into its transfer function, i.e. z-transform. The first step involves taking the Fourier Transform⁸ of all the terms in (14.43). Then we use the linearity property to pull the transform inside the summation and the time-shifting property of the z-transform to change the time-shifting terms to exponentials. Once this is done, we arrive at the following equation: $a_0 = 1$.

$$Y(z) = - \left(\sum_{k=1}^N (a_k Y(z) z^{-k}) \right) + \sum_{k=0}^M (b_k X(z) z^{-k}) \quad (14.45)$$

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} \\ &= \frac{\sum_{k=0}^M (b_k z^{-k})}{1 + \sum_{k=1}^N (a_k z^{-k})} \end{aligned} \quad (14.46)$$

14.6.2.3 Conversion to Frequency Response

Once the z-transform has been calculated from the difference equation, we can go one step further to define the frequency response of the system, or filter, that is being represented by the difference equation.

NOTE: Remember that the reason we are dealing with these formulas is to be able to aid us in filter design. A LCCDE is one of the easiest ways to represent FIR filters. By being able to find the frequency response, we will be able to look at the basic properties of any filter represented by a simple LCCDE.

Below is the general formula for the frequency response of a z-transform. The conversion is simple a matter of taking the z-transform formula, $H(z)$, and replacing every instance of z with e^{jw} .

$$\begin{aligned} H(w) &= H(z) \Big|_{z=e^{jw}} \\ &= \frac{\sum_{k=0}^M (b_k e^{-(jwk)})}{\sum_{k=0}^N (a_k e^{-(jwk)})} \end{aligned} \quad (14.47)$$

Once you understand the derivation of this formula, look at the module concerning Filter Design from the Z-Transform (Section 14.8) for a look into how all of these ideas of the Z-transform (Section 14.1), Difference Equation, and Pole/Zero Plots (Section 14.7) play a role in filter design.

14.6.3 Example

Example 14.10: Finding Difference Equation

Below is a basic example showing the opposite of the steps above: given a transfer function one can easily calculate the systems difference equation.

$$H(z) = \frac{(z+1)^2}{(z-\frac{1}{2})(z+\frac{3}{4})} \quad (14.48)$$

Given this transfer function of a time-domain filter, we want to find the difference equation. To begin with, expand both polynomials and divide them by the highest order z .

$$\begin{aligned} H(z) &= \frac{(z+1)(z+1)}{(z-\frac{1}{2})(z+\frac{3}{4})} \\ &= \frac{z^2+2z+1}{z^2+2z+1-\frac{3}{8}} \\ &= \frac{1+2z^{-1}+z^{-2}}{1+\frac{1}{4}z^{-1}-\frac{3}{8}z^{-2}} \end{aligned} \quad (14.49)$$

⁸"Derivation of the Fourier Transform" <<http://cnx.org/content/m0046/latest/>>

From this transfer function, the coefficients of the two polynomials will be our a_k and b_k values found in the general difference equation formula, (14.43). Using these coefficients and the above form of the transfer function, we can easily write the difference equation:

$$x[n] + 2x[n-1] + x[n-2] = y[n] + \frac{1}{4}y[n-1] - \frac{3}{8}y[n-2] \quad (14.50)$$

In our final step, we can rewrite the difference equation in its more common form showing the recursive nature of the system.

$$y[n] = x[n] + 2x[n-1] + x[n-2] + \frac{-1}{4}y[n-1] + \frac{3}{8}y[n-2] \quad (14.51)$$

14.6.4 Solving a LCCDE

In order for a linear constant-coefficient difference equation to be useful in analyzing a LTI system, we must be able to find the systems output based upon a known input, $x(n)$, and a set of initial conditions. Two common methods exist for solving a LCCDE: the **direct method** and the **indirect method**, the later being based on the z-transform. Below we will briefly discuss the formulas for solving a LCCDE using each of these methods.

14.6.4.1 Direct Method

The final solution to the output based on the direct method is the sum of two parts, expressed in the following equation:

$$y(n) = y_h(n) + y_p(n) \quad (14.52)$$

The first part, $y_h(n)$, is referred to as the **homogeneous solution** and the second part, $y_p(n)$, is referred to as **particular solution**. The following method is very similar to that used to solve many differential equations, so if you have taken a differential calculus course or used differential equations before then this should seem very familiar.

14.6.4.1.1 Homogeneous Solution

We begin by assuming that the input is zero, $x(n) = 0$. Now we simply need to solve the homogeneous difference equation:

$$\sum_{k=0}^N (a_k y[n-k]) = 0 \quad (14.53)$$

In order to solve this, we will make the assumption that the solution is in the form of an exponential. We will use lambda, λ , to represent our exponential terms. We now have to solve the following equation:

$$\sum_{k=0}^N (a_k \lambda^{n-k}) = 0 \quad (14.54)$$

We can expand this equation out and factor out all of the lambda terms. This will give us a large polynomial in parenthesis, which is referred to as the **characteristic polynomial**. The roots of this polynomial will be the key to solving the homogeneous equation. If there are all distinct roots, then the general solution to the equation will be as follows:

$$y_h(n) = C_1(\lambda_1)^n + C_2(\lambda_2)^n + \cdots + C_N(\lambda_N)^n \quad (14.55)$$

However, if the characteristic equation contains multiple roots then the above general solution will be slightly different. Below we have the modified version for an equation where λ_1 has K multiple roots:

$$y_h(n) = C_1(\lambda_1)^n + C_1 n(\lambda_1)^n + C_1 n^2(\lambda_1)^n + \cdots + C_1 n^{K-1}(\lambda_1)^n + C_2(\lambda_2)^n + \cdots + C_N(\lambda_N)^n \quad (14.56)$$

14.6.4.1.2 Particular Solution

The particular solution, $y_p(n)$, will be any solution that will solve the general difference equation:

$$\sum_{k=0}^N (a_k y_p(n-k)) = \sum_{k=0}^M (b_k x(n-k)) \quad (14.57)$$

In order to solve, our guess for the solution to $y_p(n)$ will take on the form of the input, $x(n)$. After guessing at a solution to the above equation involving the particular solution, one only needs to plug the solution into the difference equation and solve it out.

14.6.4.2 Indirect Method

The indirect method utilizes the relationship between the difference equation and z-transform, discussed earlier (Section 14.6.2: General Formulas from the Difference Equation), to find a solution. The basic idea is to convert the difference equation into a z-transform, as described above (Section 14.6.2.2: Conversion to Z-Transform), to get the resulting output, $Y(z)$. Then by inverse transforming this and using partial-fraction expansion, we can arrive at the solution.

14.7 Understanding Pole/Zero Plots on the Z-Plane⁹

14.7.1 Introduction to Poles and Zeros of the Z-Transform

Once the Z-transform of a system has been determined, one can use the information contained in function's polynomials to graphically represent the function and easily observe many defining characteristics. The Z-transform will have the below structure, based on Rational Functions (Section 14.5):

$$X(z) = \frac{P(z)}{Q(z)} \quad (14.58)$$

The two polynomials, $P(z)$ and $Q(z)$, allow us to find the poles and zeros (Section 13.6) of the Z-Transform.

Definition 18: zeros

1. The value(s) for z where $P(z) = 0$.
2. The complex frequencies that make the overall gain of the filter transfer function zero.

Definition 19: poles

1. The value(s) for z where $Q(z) = 0$.
2. The complex frequencies that make the overall gain of the filter transfer function infinite.

Example 14.11

Below is a simple transfer function with the poles and zeros shown below it.

$$H(z) = \frac{z+1}{\left(z-\frac{1}{2}\right)\left(z+\frac{3}{4}\right)}$$

⁹This content is available online at <<http://cnx.org/content/m10556/2.8/>>.

The zeros are: $\{-1\}$
 The poles are: $\{\frac{1}{2}, -(\frac{3}{4})\}$

14.7.2 The Z-Plane

Once the poles and zeros have been found for a given Z-Transform, they can be plotted onto the Z-Plane. The Z-plane is a complex plane with an imaginary and real axis referring to the complex-valued variable z . The position on the complex plane is given by $re^{j\theta}$ and the angle from the positive, real axis around the plane is denoted by θ . When mapping poles and zeros onto the plane, poles are denoted by an "x" and zeros by an "o". The below figure shows the Z-Plane, and examples of plotting zeros and poles onto the plane can be found in the following section.

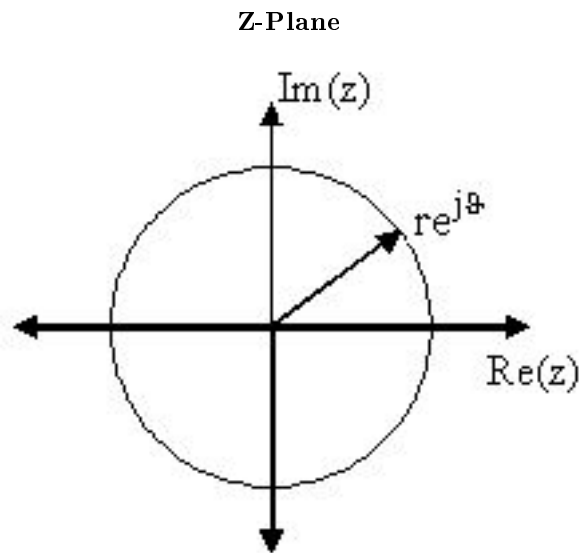


Figure 14.18

14.7.3 Examples of Pole/Zero Plots

This section lists several examples of finding the poles and zeros of a transfer function and then plotting them onto the Z-Plane.

Example 14.12: Simple Pole/Zero Plot

$$H(z) = \frac{z}{(z - \frac{1}{2})(z + \frac{3}{4})}$$

The zeros are: $\{0\}$
 The poles are: $\{\frac{1}{2}, -(\frac{3}{4})\}$

Pole/Zero Plot

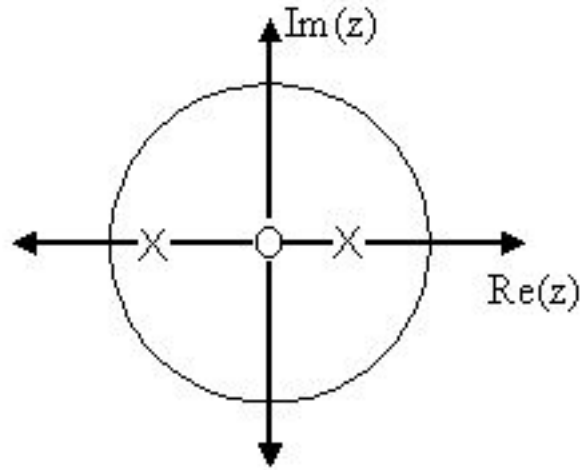


Figure 14.19: Using the zeros and poles found from the transfer function, the one zero is mapped to zero and the two poles are placed at $\frac{1}{2}$ and $-\left(\frac{3}{4}\right)$

Example 14.13: Complex Pole/Zero Plot

$$H(z) = \frac{(z-j)(z+j)}{\left(z - \left(\frac{1}{2} - \frac{1}{2}j\right)\right)\left(z - \left(\frac{1}{2} + \frac{1}{2}j\right)\right)}$$

The zeros are: $\{j, -j\}$

The poles are: $\left\{-1, \frac{1}{2} + \frac{1}{2}j, \frac{1}{2} - \frac{1}{2}j\right\}$

Pole/Zero Plot

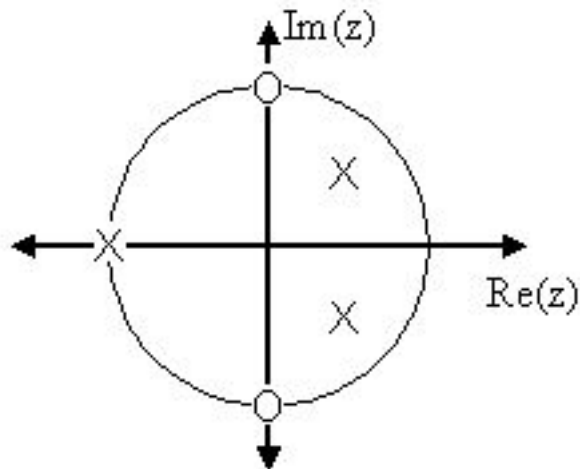


Figure 14.20: Using the zeros and poles found from the transfer function, the zeros are mapped to $\pm j$, and the poles are placed at -1 , $\frac{1}{2} + \frac{1}{2}j$ and $\frac{1}{2} - \frac{1}{2}j$

MATLAB - If access to MATLAB is readily available, then you can use its functions to easily create pole/zero plots. Below is a short program that plots the poles and zeros from the above example onto the Z-Plane.

```
% Set up vector for zeros
z = [j ; -j];

% Set up vector for poles
p = [-1 ; .5+.5j ; .5-.5j];

figure(1);
zplane(z,p);
title('Pole/Zero Plot for Complex Pole/Zero Plot Example');
```

14.7.4 Pole/Zero Plot and Region of Convergence

The region of convergence (ROC) for $X(z)$ in the complex Z-plane can be determined from the pole/zero plot. Although several regions of convergence may be possible, where each one corresponds to a different impulse response, there are some choices that are more practical. A ROC can be chosen to make the transfer function causal and/or stable depending on the pole/zero plot.

Filter Properties from ROC

- If the ROC extends outward from the outermost pole, then the system is **causal**.
- If the ROC includes the unit circle, then the system is **stable**.

Below is a pole/zero plot with a possible ROC of the Z-transform in the Simple Pole/Zero Plot (Example 14.12: Simple Pole/Zero Plot) discussed earlier. The shaded region indicates the ROC chosen for the filter. From this figure, we can see that the filter will be both causal and stable since the above listed conditions are both met.

Example 14.14

$$H(z) = \frac{z}{\left(z - \frac{1}{2}\right)\left(z + \frac{3}{4}\right)}$$

Region of Convergence for the Pole/Zero Plot

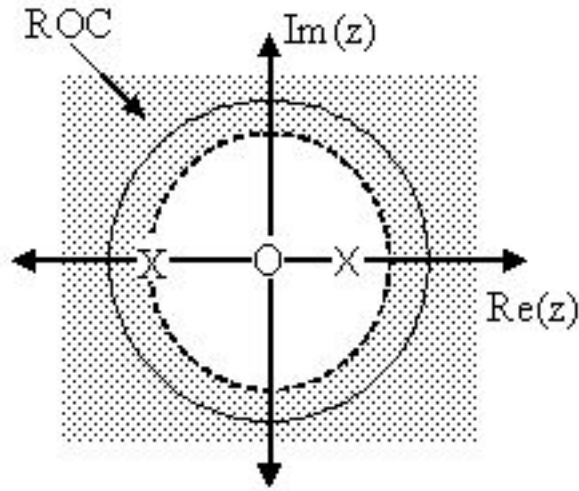


Figure 14.21: The shaded area represents the chosen ROC for the transfer function.

14.7.5 Frequency Response and the Z-Plane

The reason it is helpful to understand and create these pole/zero plots is due to their ability to help us easily design a filter. Based on the location of the poles and zeros, the magnitude response of the filter can be quickly understood. Also, by starting with the pole/zero plot, one can design a filter and obtain its transfer function very easily. Refer to this module (Section 14.8) for information on the relationship between the pole/zero plot and the frequency response.

14.8 Filter Design using the Pole/Zero Plot of a Z-Transform¹⁰

14.8.1 Estimating Frequency Response from Z-Plane

One of the motivating factors for analyzing the pole/zero plots is due to their relationship to the frequency response of the system. Based on the position of the poles and zeros, one can quickly determine the frequency response. This is a result of the correspondence between the frequency response and the transfer function evaluated on the unit circle in the pole/zero plots. The frequency response, or DTFT, of the system is defined as:

$$\begin{aligned} H(w) &= H(z) \Big|_{z=e^{jw}} \\ &= \frac{\sum_{k=0}^M (b_k e^{-jwk})}{\sum_{k=0}^N (a_k e^{-jwk})} \end{aligned} \quad (14.59)$$

Next, by factoring the transfer function into poles and zeros and multiplying the numerator and denominator by e^{jw} we arrive at the following equations:

$$H(w) = \left| \frac{b_0}{a_0} \right| \frac{\prod_{k=1}^M (|e^{jw} - c_k|)}{\prod_{k=1}^N (|e^{jw} - d_k|)} \quad (14.60)$$

¹⁰This content is available online at <<http://cnx.org/content/m10548/2.9/>>.

From (14.60) we have the frequency response in a form that can be used to interpret physical characteristics about the filter's frequency response. The numerator and denominator contain a product of terms of the form $|e^{jw} - h|$, where h is either a zero, denoted by c_k or a pole, denoted by d_k . Vectors are commonly used to represent the term and its parts on the complex plane. The pole or zero, h , is a vector from the origin to its location anywhere on the complex plane and e^{jw} is a vector from the origin to its location on the unit circle. The vector connecting these two points, $|e^{jw} - h|$, connects the pole or zero location to a place on the unit circle dependent on the value of w . From this, we can begin to understand how the magnitude of the frequency response is a ratio of the distances to the poles and zero present in the z-plane as w goes from zero to π . These characteristics allow us to interpret $|H(w)|$ as follows:

$$|H(w)| = \left| \frac{b_0}{a_0} \right| \frac{\prod \text{"distances from zeros"}}{\prod \text{"distances from poles"}} \quad (14.61)$$

In conclusion, using the distances from the unit circle to the poles and zeros, we can plot the frequency response of the system. As w goes from 0 to 2π , the following two properties, taken from the above equations, specify how one should draw $|H(w)|$.

While moving around the unit circle...

1. if close to a zero, then the magnitude is small. If a zero is on the unit circle, then the frequency response is zero at that point.
2. if close to a pole, then the magnitude is large. If a pole is on the unit circle, then the frequency response goes to infinity at that point.

14.8.2 Drawing Frequency Response from Pole/Zero Plot

Let us now look at several examples of determining the magnitude of the frequency response from the pole/zero plot of a z-transform. If you have forgotten or are unfamiliar with pole/zero plots, please refer back to the Pole/Zero Plots (Section 14.7) module.

Example 14.15

In this first example we will take a look at the very simple z-transform shown below:

$$H(z) = z + 1 = 1 + z^{-1}$$

$$H(w) = 1 + e^{-(jw)}$$

For this example, some of the vectors represented by $|e^{jw} - h|$, for random values of w , are explicitly drawn onto the complex plane shown in the figure below. These vectors show how the amplitude of the frequency response changes as w goes from 0 to 2π , and also show the physical meaning of the terms in (14.60) above. One can see that when $w = 0$, the vector is the longest and thus the frequency response will have its largest amplitude here. As w approaches π , the length of the vectors decrease as does the amplitude of $|H(w)|$. Since there are no poles in the transform, there is only this one vector term rather than a ratio as seen in (14.60).

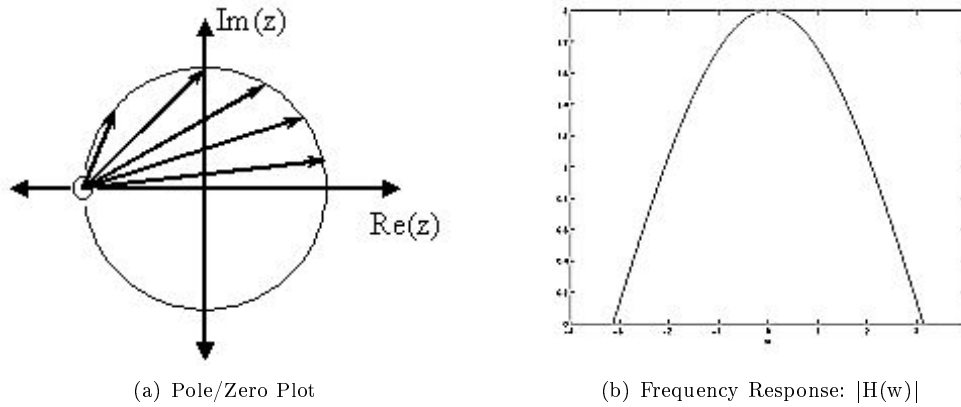


Figure 14.22: The first figure represents the pole/zero plot with a few representative vectors graphed while the second shows the frequency response with a peak at $+2$ and graphed between plus and minus π .

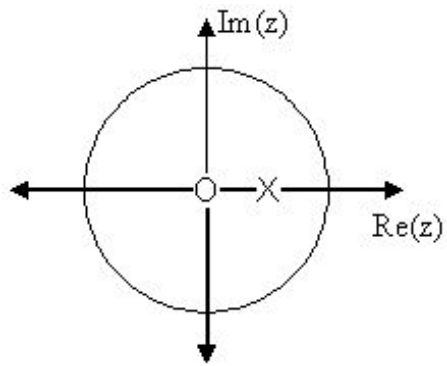
Example 14.16

For this example, a more complex transfer function is analyzed in order to represent the system's frequency response.

$$H(z) = \frac{z}{z - \frac{1}{2}} = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

$$H(w) = \frac{1}{1 - \frac{1}{2}e^{-jw}}$$

Below we can see the two figures described by the above equations. The Figure 14.23(a) (Pole/Zero Plot) represents the basic pole/zero plot of the z -transform, $H(w)$. Figure 14.23(b) (Frequency Response: $|H(w)|$) shows the magnitude of the frequency response. From the formulas and statements in the previous section, we can see that when $w = 0$ the frequency will peak since it is at this value of w that the pole is closest to the unit circle. The ratio from (14.60) helps us see the mathematics behind this conclusion and the relationship between the distances from the unit circle and the poles and zeros. As w moves from 0 to π , we see how the zero begins to mask the effects of the pole and thus force the frequency response closer to 0.



(a) Pole/Zero Plot

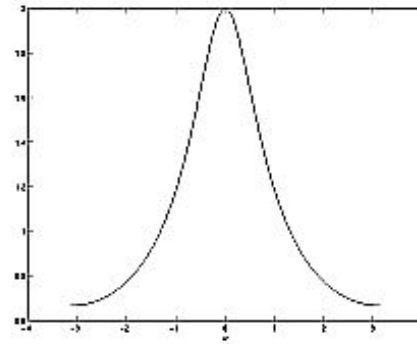
(b) Frequency Response: $|H(w)|$

Figure 14.23: The first figure represents the pole/zero plot while the second shows the frequency response with a peak at $+2$ and graphed between plus and minus π .

Chapter 15

Appendix: Hilbert Spaces and Orthogonal Expansions

15.1 Vector Spaces¹

Introduction

Definition 20: Vector space

A linear vector space S is a collection of "vectors" such that (1) if $f_1 \in S \Rightarrow \alpha f_1 \in S$ for all scalars α (where $\alpha \in \mathbb{R}$ or $\alpha \in \mathbb{C}$) and (2) if $f_1 \in S, f_2 \in S$, then $f_1 + f_2 \in S$

To define an abstract linear vector space, we need

- A set of things called "vectors" (X)
- A set of things called "scalars" (A)
- A vector addition operator (+)
- A scalar multiplication operator (*)

The operators need to have all the properties of given below. Closure is usually the most important to show.

15.1.2 Vector Spaces

If the scalars α are real, S is called a **real vector space**.

If the scalars α are complex, S is called a **complex vector space**.

If the "vectors" in S are functions of a continuous variable, we sometimes call S a **linear function space**

15.1.2.1 Properties

We define a set V to be a vector space if

1. $\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$ for each \mathbf{x} and \mathbf{y} in V
2. $\mathbf{x} + (\mathbf{y} + \mathbf{z}) = (\mathbf{x} + \mathbf{y}) + \mathbf{z}$ for each \mathbf{x}, \mathbf{y} , and \mathbf{z} in V
3. There is a unique "zero vector" such that $\mathbf{x} + 0 = \mathbf{x}$ for each \mathbf{x} in V
4. For each \mathbf{x} in V there is a unique vector $-\mathbf{x}$ such that $\mathbf{x} + (-\mathbf{x}) = 0$.
5. $1\mathbf{x} = \mathbf{x}$
6. $(c_1 c_2) \mathbf{x} = c_1 (c_2 \mathbf{x})$ for each \mathbf{x} in V and c_1 and c_2 in \mathbb{C} .
7. $c(\mathbf{x} + \mathbf{y}) = c\mathbf{x} + c\mathbf{y}$ for each \mathbf{x} and \mathbf{y} in V and c in \mathbb{C} .
8. $(c_1 + c_2) \mathbf{x} = c_1 \mathbf{x} + c_2 \mathbf{x}$ for each \mathbf{x} in V and c_1 and c_2 in \mathbb{C} .

¹This content is available online at <<http://cnx.org/content/m10767/2.4/>>.

15.1.2.2 Examples

- $\mathbb{R}^n = \text{real vector space}$
- $\mathbb{C}^n = \text{complex vector space}$
- $L^1(\mathbb{R}) = \left\{ f(t) \mid \int_{-\infty}^{\infty} |f(t)| dt < \infty \right\}$ is a vector space
- $L^\infty(\mathbb{R}) = \{ f(t) \mid f(t) \text{ is bounded} \}$ is a vector space
- $L^2(\mathbb{R}) = \left\{ f(t) \mid \int_{-\infty}^{\infty} (|f(t)|)^2 dt < \infty \right\} = \text{finite energy signals}$ is a vector space
- $L^2([0, T]) = \text{finite energy functions on interval } [0, T]$
- $\ell^1(\mathbb{Z}), \ell^2(\mathbb{Z}), \ell^\infty(\mathbb{Z})$ are vector spaces
- The collection of functions piecewise constant between the integers is a vector space

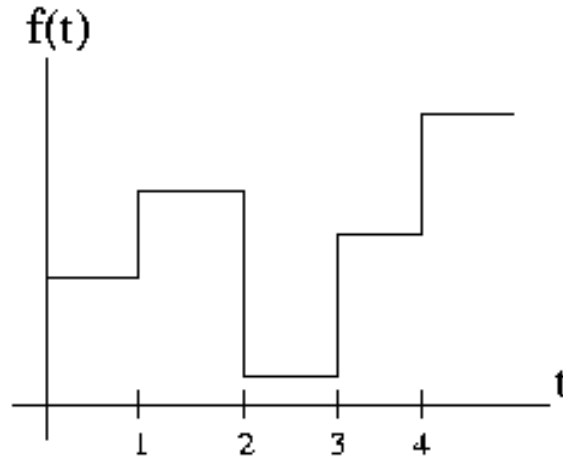


Figure 15.1

- $\mathbb{R}_+^2 = \left\{ \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \mid x_0 > 0 \text{ and } x_1 > 0 \right\}$ is not a vector space. $\begin{pmatrix} 1 \\ 1 \end{pmatrix} \in \mathbb{R}_+^2$, but $\alpha \begin{pmatrix} 1 \\ 1 \end{pmatrix} \notin \mathbb{R}_+^2$, $\alpha < 0$
- $D = \{z \in \mathbb{C}, |z| \leq 1\}$ is not a vector space. $z_1 = 1 \in D$, $z_2 = j \in D$, but $z_1 + z_2 \notin D$, $|z_1 + z_2| = \sqrt{2} > 1$

NOTE: Vector spaces can be collections of functions, collections of sequences, as well as collections of traditional vectors (i.e. finite lists of numbers)

15.2 Norms²

15.2.1 Introduction

Much of the language in this section will be familiar to you - you should have previously been exposed to the concepts of

- inner products (Section 15.3)
- orthogonality
- basis expansions (Section 15.8)

in the context of \mathbb{R}^n . We're going to take what we know about vectors and apply it to functions (continuous time signals).

15.2.2 Norms

The **norm** of a vector is a real number that represents the "size" of the vector.

Example 15.1

In \mathbb{R}^2 , we can define a norm to be a vectors geometric length.

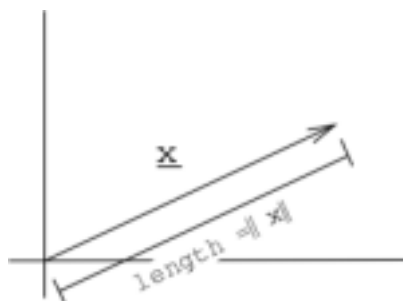


Figure 15.2

$$\mathbf{x} = (x_0, x_1)^T, \text{ norm } \|\mathbf{x}\| = \sqrt{x_0^2 + x_1^2}$$

Mathematically, a norm $\|\cdot\|$ is just a function (taking a vector and returning a real number) that satisfies three rules.

To be a norm, $\|\cdot\|$ must satisfy:

1. the norm of every vector is positive $\|\mathbf{x}\| > 0$, $\mathbf{x} \in S$
2. scaling a vector scales the norm by the same amount $\|\alpha\mathbf{x}\| = |\alpha| \|\mathbf{x}\|$ for all vectors \mathbf{x} and scalars α
3. Triangle Property: $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all vectors \mathbf{x}, \mathbf{y} . "The "size" of the sum of two vectors is less than or equal to the sum of their sizes"

A vector space (Section 15.1) with a well defined norm is called a **normed vector space** or **normed linear space**.

²This content is available online at <http://cnx.org/content/m10768/2.4/>.

15.2.2.1 Examples

Example 15.2

\mathbb{R}^n (or \mathbb{C}^n), $\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{n-1} \end{pmatrix}$, $\|x\|_1 = \sum_{i=0}^{n-1} (|x_i|)$, \mathbb{R}^n with this norm is called $\ell^1([0, n-1])$.

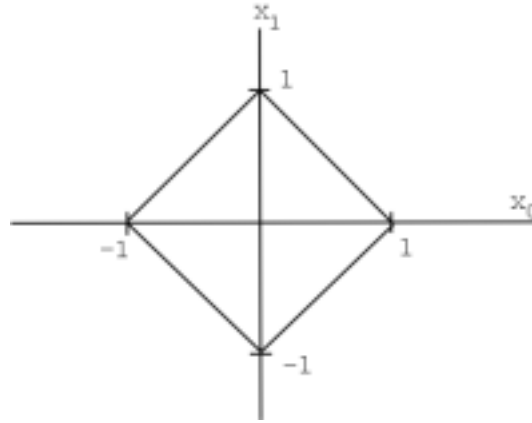


Figure 15.3: Collection of all $\mathbf{x} \in \mathbb{R}^2$ with $\|x\|_1 = 1$

Example 15.3

\mathbb{R}^n (or \mathbb{C}^n), with norm $\|x\|_2 = \left(\sum_{i=0}^{n-1} (|x_i|^2) \right)^{\frac{1}{2}}$, \mathbb{R}^n is called $\ell^2([0, n-1])$ (the usual "Euclidean" norm).

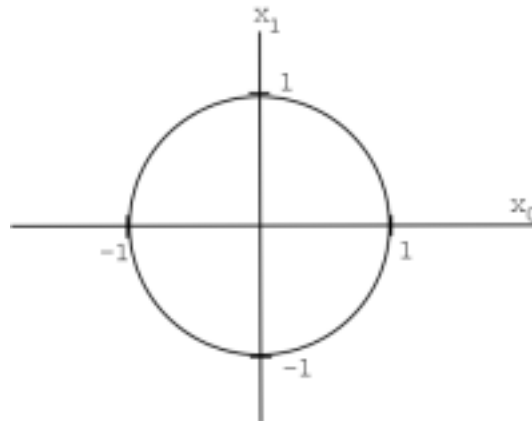


Figure 15.4: Collection of all $\mathbf{x} \in \mathbb{R}^2$ with $\|x\|_2 = 1$

Example 15.4

\mathbb{R}^n (or \mathbb{C}^n , with norm $\|x\|_\infty = \max_i \{|x_i|\}$) is called $\ell^\infty([0, n-1])$

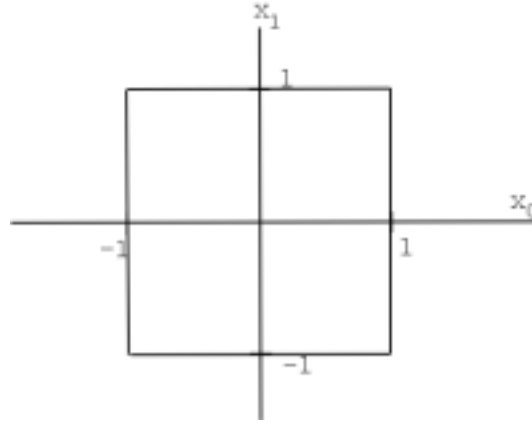


Figure 15.5: $x \in \mathbb{R}^2$ with $\|x\|_\infty = 1$

15.2.2.2 Spaces of Sequences and Functions

We can define similar norms for spaces of sequences and functions.

Discrete time signals = sequences of numbers

$$x[n] = \{\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots\}$$

- $\|x(n)\|_1 = \sum_{i=-\infty}^{\infty} (|x[i]|)$, $x[n] \in \ell^1(\mathbb{Z}) \Rightarrow \|x\|_1 < \infty$
- $\|x(n)\|_2 = \left(\sum_{i=-\infty}^{\infty} (|x[i]|^2)\right)^{\frac{1}{2}}$, $x[n] \in \ell^2(\mathbb{Z}) \Rightarrow \|x\|_2 < \infty$
- $\|x(n)\|_p = \left(\sum_{i=-\infty}^{\infty} (|x[i]|^p)\right)^{\frac{1}{p}}$, $x[n] \in \ell^p(\mathbb{Z}) \Rightarrow \|x\|_p < \infty$
- $\|x(n)\|_\infty = \sup_i |x[i]|$, $x[n] \in \ell^\infty(\mathbb{Z}) \Rightarrow \|x\|_\infty < \infty$

For continuous time functions:

- $\|f(t)\|_p = \left(\int_{-\infty}^{\infty} (|f(t)|^p dt)\right)^{\frac{1}{p}}$, $f(t) \in L^p(\mathbb{R}) \Rightarrow \|f(t)\|_p < \infty$
- (On the interval) $\|f(t)\|_p = \left(\int_0^T (|f(t)|^p dt)\right)^{\frac{1}{p}}$, $f(t) \in L^p([0, T]) \Rightarrow \|f(t)\|_p < \infty$

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10768/latest/NormCalc.llb>

15.3 Inner Products³

15.3.1 Definition: Inner Product

You may have run across **inner products**, also called **dot products**, on \mathbb{R}^n before in some of your math or science courses. If not, we define the inner product as follows, given we have some $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$

Definition 21: inner product

The inner product is defined mathematically as:

$$\begin{aligned}
 \langle \mathbf{x}, \mathbf{y} \rangle &= \mathbf{y}^T \mathbf{x} \\
 &= \begin{pmatrix} y_0 & y_1 & \cdots & y_{n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} \\
 &= \sum_{i=0}^{n-1} (x_i y_i)
 \end{aligned} \tag{15.1}$$

15.3.1.1 Inner Product in 2-D

If we have $\mathbf{x} \in \mathbb{R}^2$ and $\mathbf{y} \in \mathbb{R}^2$, then we can write the inner product as

$$\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\| \|\mathbf{y}\| \cos(\theta) \tag{15.2}$$

where θ is the angle between \mathbf{x} and \mathbf{y} .

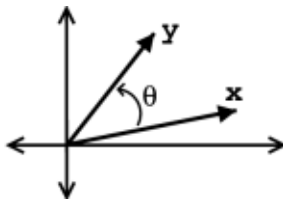


Figure 15.6: General plot of vectors and angle referred to in above equations.

Geometrically, the inner product tells us about the *strength* of \mathbf{x} in the direction of \mathbf{y} .

Example 15.5

For example, if $\|\mathbf{x}\| = 1$, then

$$\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{y}\| \cos(\theta)$$

³This content is available online at <http://cnx.org/content/m10755/2.6/>.

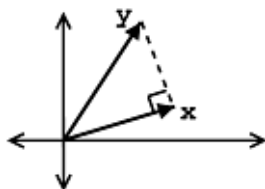


Figure 15.7: Plot of two vectors from above example.

The following characteristics are revealed by the inner product:

- $\langle \mathbf{x}, \mathbf{y} \rangle$ measures the length of the **projection** of \mathbf{y} onto \mathbf{x} .
- $\langle \mathbf{x}, \mathbf{y} \rangle$ is **maximum** (for given $\|\mathbf{x}\|, \|\mathbf{y}\|$) when \mathbf{x} and \mathbf{y} are in the same direction ($\theta = 0 \Rightarrow \cos(\theta) = 1$).
- $\langle \mathbf{x}, \mathbf{y} \rangle$ is zero when $\cos(\theta) = 0 \Rightarrow \theta = 90^\circ$, i.e. \mathbf{x} and \mathbf{y} are **orthogonal**.

15.3.1.2 Inner Product Rules

In general, an inner product on a complex vector space is just a function (taking two vectors and returning a complex number) that satisfies certain rules:

- Conjugate Symmetry:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle^*$$

- Scaling:

$$\langle \alpha \mathbf{x}, \mathbf{y} \rangle = \alpha \langle \mathbf{x}, \mathbf{y} \rangle$$

- Additivity:

$$\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$$

- "Positivity":

$$\langle \mathbf{x}, \mathbf{x} \rangle \geq 0, \quad \mathbf{x} \neq \mathbf{0}$$

Definition 22: orthogonal

We say that \mathbf{x} and \mathbf{y} are orthogonal if:

$$\langle \mathbf{x}, \mathbf{y} \rangle = 0$$

15.4 Hilbert Spaces⁴

15.4.1 Hilbert Spaces

A vector space S with a valid inner product (Section 15.3) defined on it is called an **inner product space**, which is also a **normed linear space**. A **Hilbert space** is an inner product space that is complete with respect to the norm defined using the inner product. Hilbert spaces are named after David Hilbert⁵, who developed this idea through his studies of integral equations. We define our valid norm using the inner product as:

$$\| \mathbf{x} \| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} \quad (15.3)$$

Hilbert spaces are useful in studying and generalizing the concepts of Fourier expansion, Fourier transforms, and are very important to the study of quantum mechanics. Hilbert spaces are studied under the functional analysis branch of mathematics.

15.4.1.1 Examples of Hilbert Spaces

Below we will list a few examples of Hilbert spaces⁶. You can verify that these are valid inner products at home.

- For \mathbb{C}^n ,

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^T \mathbf{x} = \begin{pmatrix} y_0^* & y_1^* & \dots & y_{n-1}^* \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} = \sum_{i=0}^{n-1} (x_i y_i^*)$$

- Space of finite energy complex functions: $L^2(\mathbb{R})$

$$\langle \mathbf{f}, \mathbf{g} \rangle = \int_{-\infty}^{\infty} f(t) g(t)^* dt$$

- Space of square-summable sequences: $\ell^2(\mathbb{Z})$

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=-\infty}^{\infty} (x[i] y[i]^*)$$

15.5 Cauchy-Schwarz Inequality⁷

15.5.1 Introduction

Recall in \mathbb{R}^2 , $\langle \mathbf{x}, \mathbf{y} \rangle = \| \mathbf{x} \| \| \mathbf{y} \| \cos(\theta)$

$$| \langle \mathbf{x}, \mathbf{y} \rangle | \leq \| \mathbf{x} \| \| \mathbf{y} \| \quad (15.4)$$

The same relation holds for inner product spaces (Section 15.3) in general...

⁴This content is available online at <http://cnx.org/content/m10840/2.4/>.

⁵<http://www-history.mcs.st-andrews.ac.uk/history/Mathematicians/Hilbert.html>

⁶"Hilbert Spaces" <http://cnx.org/content/m10434/latest/>

⁷This content is available online at <http://cnx.org/content/m10757/2.5/>.

15.5.1.1 Cauchy-Schwarz Inequality

Definition 23: Cauchy-Schwarz Inequality

For \mathbf{x}, \mathbf{y} in an inner product space

$$| \langle \mathbf{x}, \mathbf{y} \rangle | \leq \| \mathbf{x} \| \| \mathbf{y} \|$$

with equality holding *if and only if* \mathbf{x} and \mathbf{y} are linearly dependent (Section 5.1.1: Linear Independence), *i.e.* $\mathbf{x} = \alpha \mathbf{y}$ for some scalar α .

15.5.2 Matched Filter Detector

Also referred to as Cauchy-Schwarz's "Killer App."

15.5.2.1 Concept behind Matched Filter

If we are given two vectors, \mathbf{f} and \mathbf{g} , then the Cauchy-Schwarz Inequality (CSI) is *maximized* when $\mathbf{f} = \alpha \mathbf{g}$. This tells us:

- \mathbf{f} is in the same "direction" as \mathbf{g}
- if \mathbf{f} and \mathbf{g} are functions, $\mathbf{f} = \alpha \mathbf{g}$ means \mathbf{f} and \mathbf{g} have the same shape.

For example, say we are in a situation where we have a set of signals, defined as $\{g_1(t), g_2(t), \dots, g_k(t)\}$, and we want to be able to tell which, if any, of these signals resemble another given signal $f(t)$.

STRATEGY: In order to find the signal(s) that resembles $f(t)$ we will take the inner products. If $g_i(t)$ resembles $f(t)$, then the absolute value of the inner product, $| \langle f(t), g_i(t) \rangle |$, will be *large*.

This idea of being able to measure and rank the "likeness" of two signals leads us to the **Matched Filter Detector**.

15.5.2.2 Comparing Signals

The simplest use of the Matched Filter would be to take a set of "candidate" signals, say our set of $\{g_1(t), g_2(t), \dots, g_k(t)\}$, and try to match it to a "template" signal, $f(t)$. For example say we are given the below template (Figure 15.8 (Template Signal)) and candidate signals (Figure 15.9 (Candidate Signals)):

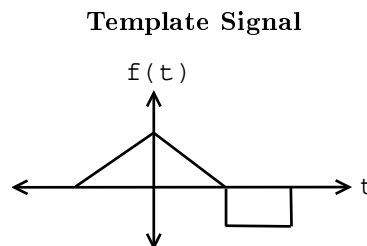


Figure 15.8: Our signal we wish to find match of.

Candidate Signals

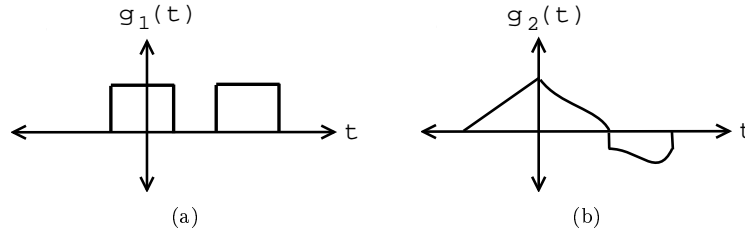


Figure 15.9: Clearly by looking at these we can see which signal will provide the better match to our template signal.

Now if our only question was which function was a closer match to $f(t)$ then we can easily come up with the answer based on inspection - $g_2(t)$. However, this will not always be the case. Also, we may want to develop a method, or algorithm, that could automate these comparisons. Or perhaps we wish to have a quantitative value expressing just how similar the signals are. To address these issues, we will lay out a more formal approach to comparing the signals, which will, as mentioned above, be based on the inner product.

In order to see which of our candidate signals, $g_1(t)$ or $g_2(t)$, best resembles $f(t)$ we need to perform the following steps:

- Normalize the $g_i(t)$
- Take the inner product with $f(t)$
- Find the biggest!

Or, putting it mathematically:

$$\text{Bestcandidate} = \underset{i}{\operatorname{argmax}} \frac{|\langle \mathbf{f}, g_i \rangle|}{\|g_i\|} \quad (15.5)$$

15.5.2.3 Finding a Pattern

Extending these thoughts of using the Matched Filter to find similarities among signals, we can use the same idea to search for a pattern in a long signal. The idea is simply to repeatedly perform the same calculation as we did previously; however, now instead of calculating on different signals we will simply perform the inner product with different shifted versions of our "pattern" signal. For example, say we have the following two signals - a pattern signal (Figure 15.10 (Pattern Signal)) and long signal (Figure 15.11 (Long Signal)).

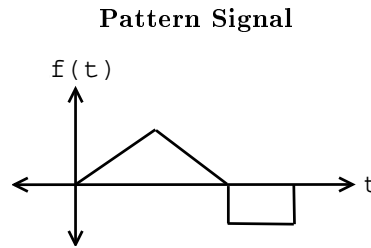


Figure 15.10: The pattern we are looking for in a our long signal having a length T .

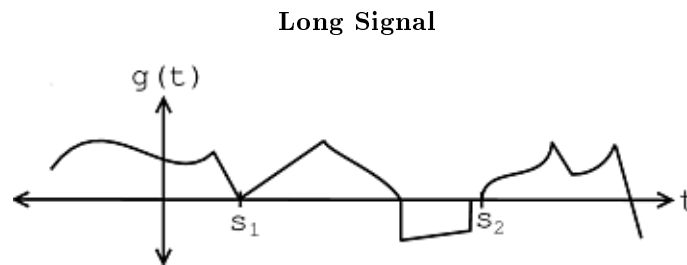


Figure 15.11: Here is the long signal that contains a piece that resembles our pattern signal.

Here we will look at two shifts of our pattern signal, shifting the signal by s_1 and s_2 . These two possibilities yield the following calculations and results:

- Shift of s_1 :

$$\frac{\int_{s_1}^{s_1+T} g(t) f(t - s_1) dt}{\sqrt{\int_{s_1}^{s_1+T} (|g(t)|)^2 dt}} = \text{"large"} \quad (15.6)$$

- Shift of s_2 :

$$\frac{\int_{s_2}^{s_2+T} g(t) f(t - s_2) dt}{\sqrt{\int_{s_2}^{s_2+T} (|g(t)|)^2 dt}} = \text{"small"} \quad (15.7)$$

Therefore, we can define a generalized equation for our matched filter:

$$m(s) = \text{matchedfilter} \quad (15.8)$$

$$m(s) = \frac{\int_s^{s+T} g(t) f(t - s) dt}{(\|g(t)\|_{L^2([s, s+T])})} \quad (15.9)$$

where the numerator in (15.9) is the convolution of $g(t) * f(-t)$. Now in order to decide whether or not the result from our matched filter detector is high enough to indicate an acceptable match between the two signals, we define some **threshold**. If

$$m(s_0) \geq \text{threshold}$$

then we have a match at location s_0 .

15.5.2.4 Practical Examples

15.5.2.4.1 Image Detection

In 2-D, this concept is used to match images together, such as verifying fingerprints for security or to match photos of someone. For example, this idea could be used for the ever-popular "Where's Waldo?" books! If we are given the below template (Figure 15.12(a)) and piece of a "Where's Waldo?" book (Figure 15.12(b)),



Figure 15.12: Example of "Where's Waldo?" picture. Our Matched Filter Detector can be implemented to find a possible match for Waldo.

then we could easily develop a program to find the closest resemblance to the image of Waldo's head in the larger picture. We would simply implement our same match filter algorithm: take the inner products at each shift and see how large our resulting answers are. This idea was implemented on this same picture for a Signals and Systems Project⁸ at Rice University (click the link to learn more).

15.5.2.4.2 Communications Systems

Matched filter detector are also commonly used in Communications Systems⁹. In fact, they are the *optimal* detectors in Gaussian noise. Signals in the real-world are often distorted by the environment around them, so there is a constant struggle to develop ways to be able to receive a distorted signal and then be able to filter it in some way to determine what the original signal was. Matched filters provide one way to compare a received signal with two possible original ("template") signals and determine which one is the closest match to the received signal.

For example, below we have a simplified example of Frequency Shift Keying¹⁰ (FSK) where we having the following coding for '1' and '0':

⁸<http://www.owl.net.rice.edu/~elec301/Projects99/waldo/process.html>

⁹"Structure of Communication Systems" <<http://cnx.org/content/m0002/latest/>>

¹⁰"Frequency Shift Keying" <<http://cnx.org/content/m0545/latest/>>

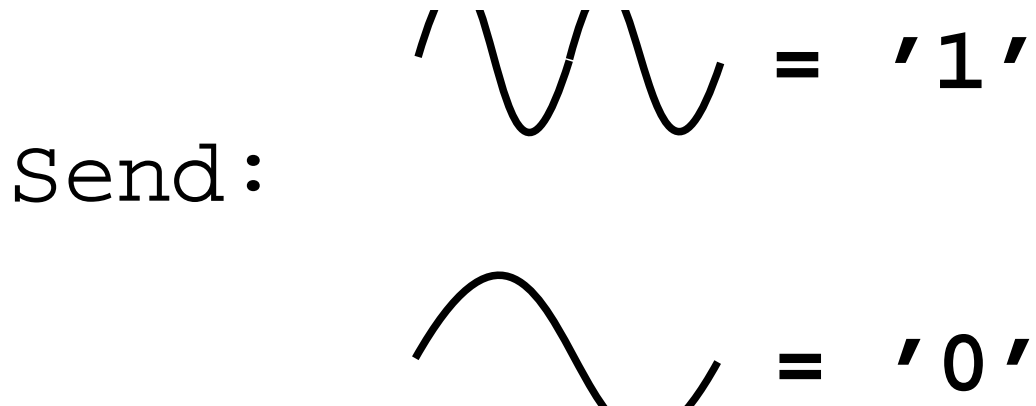


Figure 15.13: Frequency Shift Keying for '1' and '0'.

Based on the above coding, we can create digital signals based on 0's and 1's by putting together the above two "codes" in an infinite number of ways. For this example we will transmit a basic 3-bit number, 101, which is displayed in Figure 15.14:

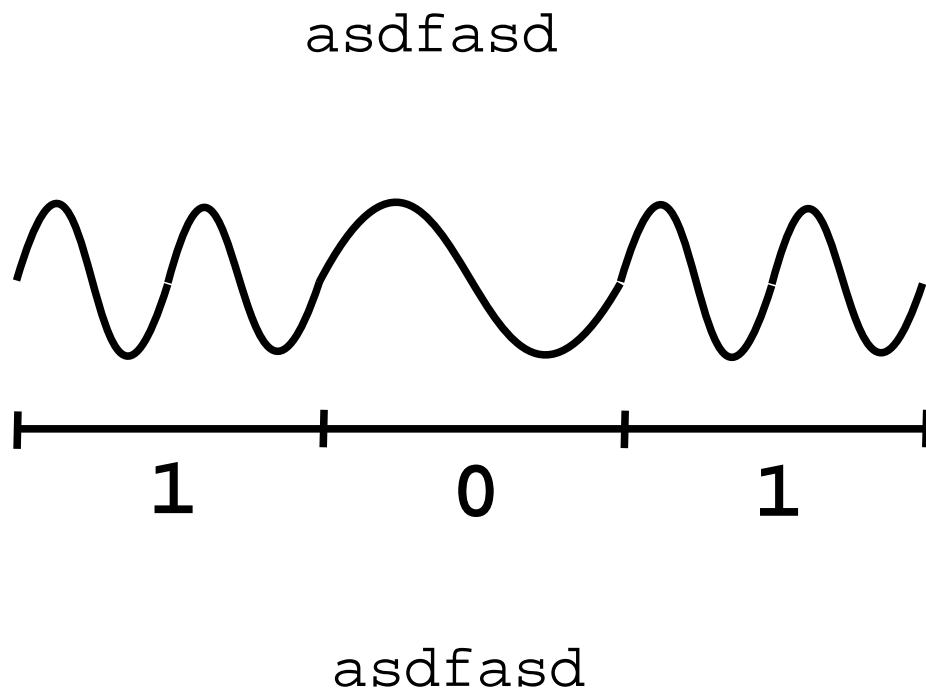


Figure 15.14: The bit stream "101" coded with the above FSK.

Now, the signal picture above represents our original signal that will be transmitted over some communication system, which will inevitably pass through the "communications channel," the part of the system that will distort and alter our signal. As long as the noise is not too great, our matched filter should keep us from having to worry about these changes to our transmitted signal. Once this signal has been received, we will pass the noisy signal through a simple system, similar to the simplified version shown in Figure 15.15:

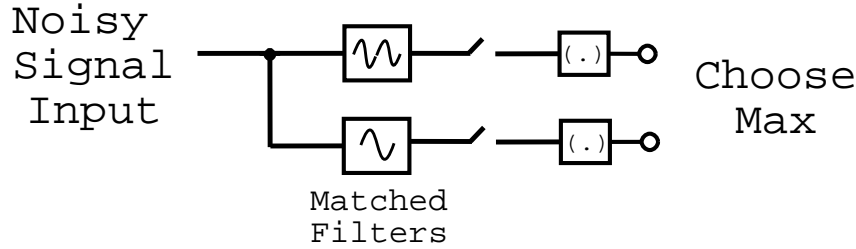


Figure 15.15: Block diagram of matched filter detector.

Figure 15.15 basically shows that our noisy signal will be passed in (we will assume that it passes in one "bit" at a time) and this signal will be split and passed to two different matched filter detectors. Each one will compare the noisy, received signal to one of the two codes we defined for '1' and '0.' Then this value will be passed on and whichever value is higher (i.e. whichever FSK code signal the noisy signal most resembles) will be the value that the receiver takes. For example, the first bit that will be sent through will be a '1' so the upper level of the block diagram will have a higher value, thus denoting that a '1' was sent by the signal, even though the signal may appear very noisy and distorted.

15.5.3 Proof of CSI

Here will look at the proof of our Cauchy-Schwarz Inequality (CSI) for a **real vector space**.

Theorem 15.1: CSI for Real Vector Space

For $f \in HilbertSpaceS$ and $g \in HilbertSpaceS$, show:

$$| \langle f, g \rangle | \leq \| f \| \| g \| \tag{15.10}$$

with equality if and only if $g = \alpha f$.

Proof:

- If $g = \alpha f$, show $| \langle f, g \rangle | = \| f \| \| g \|$

$$| \langle f, g \rangle | = | \langle f, \alpha f \rangle | = |\alpha| | \langle f, f \rangle | = |\alpha| (\| f \|^2)$$

$$| \langle f, g \rangle | = \| f \| (|\alpha| \| f \|) = \| f \| \| g \|$$

This verifies our above statement of the CSI!

- If $g \neq \alpha f$, show $| \langle f, g \rangle | < \| f \| \| g \|$ where we have $\beta f + g \neq 0$, $\beta \in \mathbb{R}$

$$\begin{aligned} 0 < (\| \beta f + g \|^2) &= \langle \beta f + g, \beta f + g \rangle = \beta^2 \langle f, f \rangle + 2\beta \langle f, g \rangle + \langle g, g \rangle \\ &= \beta^2 (\| f \|^2) + 2\beta \langle f, g \rangle + (\| g \|^2) \end{aligned}$$

And we get a quadratic in β . Visually, the quadratic polynomial in $\beta > 0$ for all β . Also, note that this polynomial has no real roots and the discriminant is less than 0. - BLAH BLAH BLAH -

$$a\beta^2 + b\beta + c$$

has discriminant $\beta^2 - 4ac$ where we have:

$$a = (\|f\|)^2$$

$$b = 2 \langle f, g \rangle$$

$$c = (\|g\|)^2$$

Therefore, we can plug these values into the above polynomial discriminant to get:

$$4(\langle f, g \rangle)^2 - 4(\|f\|)^2(\|g\|)^2 < 0 \quad (15.11)$$

$$|\langle f, g \rangle| < \|f\| \|g\| \quad (15.12)$$

And finally we have proven the Cauchy-Schwarz Inequality formula for real vector spaces.

QUESTION: What changes do we have to make to the proof for a complex vector space? (try to figure this out at home)

15.6 Common Hilbert Spaces¹¹

15.6.1 Common Hilbert Spaces

Below we will look at the four most common Hilbert spaces (Section 15.3) that you will have to deal with when discussing and manipulating signals and systems.

15.6.1.1

\mathbb{R}^n (real scalars) and \mathbb{C}^n (complex scalars), also called $\ell^2([0, n-1])$

$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{n-1} \end{pmatrix}$ is a list of numbers (finite sequence). The inner product (Section 15.3) for our two spaces

are as follows:

- Inner product \mathbb{R}^n :

$$\begin{aligned} \langle \mathbf{x}, \mathbf{y} \rangle &= \mathbf{y}^T \mathbf{x} \\ &= \sum_{i=0}^{n-1} (x_i y_i) \end{aligned} \quad (15.13)$$

- Inner product \mathbb{C}^n :

$$\begin{aligned} \langle \mathbf{x}, \mathbf{y} \rangle &= \mathbf{y}^{T*} \mathbf{x} \\ &= \sum_{i=0}^{n-1} (x_i y_i^*) \end{aligned} \quad (15.14)$$

¹¹This content is available online at <http://cnx.org/content/m10759/2.5/>.

Model for: Discrete time signals on the interval $[0, n - 1]$ or periodic (with period n) discrete time signals.

$$\begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{n-1} \end{pmatrix}$$

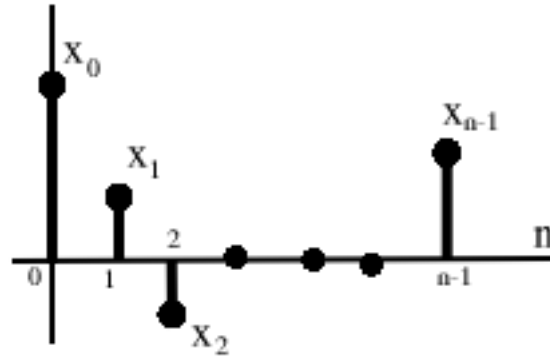


Figure 15.16

15.6.1.2

$f \in L^2([a, b])$ is a *finite energy function* on $[a, b]$

Inner Product

$$\langle f, g \rangle = \int_a^b f(t) g(t)^* dt \tag{15.15}$$

Model for: continuous time signals on the interval $[a, b]$ or periodic (with period $T = b - a$) continuous time signals

15.6.1.3

$x \in \ell^2(\mathbb{Z})$ is an *infinite sequence of numbers* that's square-summable

Inner product

$$\langle x, y \rangle = \sum_{i=-\infty}^{\infty} (x[i] y[i]^*) \tag{15.16}$$

Model for: discrete time, non-periodic signals

15.6.1.4

$f \in L^2(\mathbb{R})$ is a *finite energy function* on all of \mathbb{R} .

Inner product

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(t) g(t)^* dt \tag{15.17}$$

Model for: continuous time, non-periodic signals

15.6.2 Associated Fourier Analysis

Each of these 4 Hilbert spaces has a type of Fourier analysis associated with it.

- $L^2([a, b]) \rightarrow$ Fourier series
- $\ell^2([0, n - 1]) \rightarrow$ Discrete Fourier Transform
- $L^2(\mathbb{R}) \rightarrow$ Fourier Transform
- $\ell^2(\mathbb{Z}) \rightarrow$ Discrete Time Fourier Transform

But all 4 of these are based on the same principles (Hilbert space).

IMPORTANT NOTE: Not all normed spaces are Hilbert spaces

For example: $L^1(\mathbb{R})$, $\|f\|_1 = \int |f(t)| dt$. Try as you might, you can't find an inner product that induces this norm, i.e. a $\langle \cdot, \cdot \rangle$ such that

$$\begin{aligned} \langle f, f \rangle &= \left(\int (|f(t)|)^2 dt \right)^2 \\ &= (\|f\|_1)^2 \end{aligned} \tag{15.18}$$

In fact, of all the $L^p(\mathbb{R})$ spaces, $L^2(\mathbb{R})$ is the *only one* that is a Hilbert space.

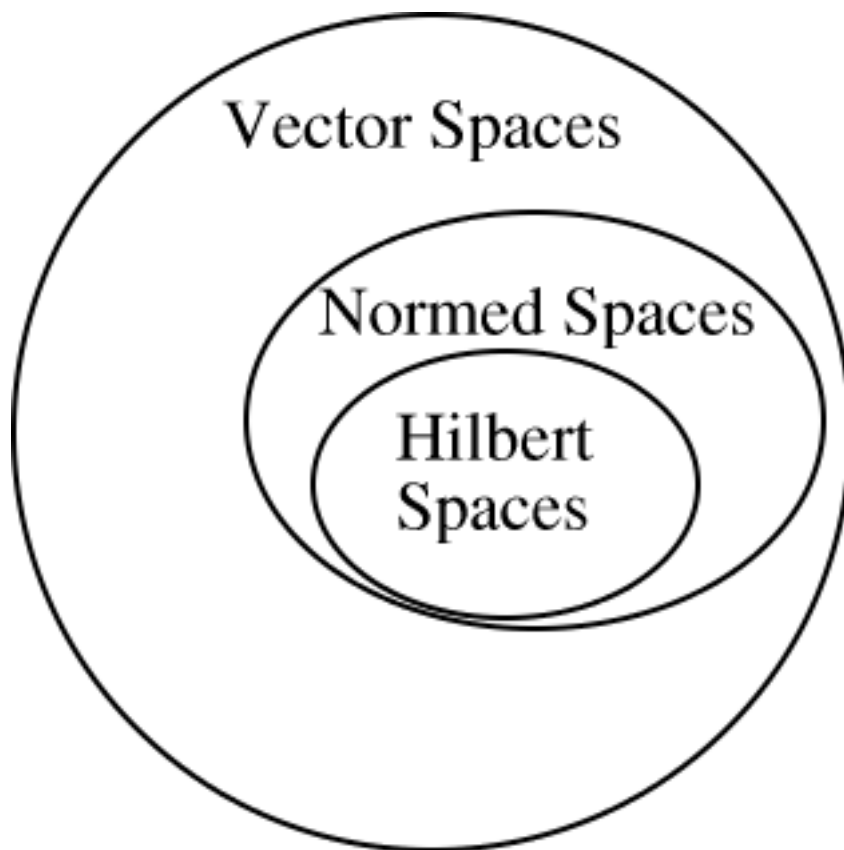


Figure 15.17

Hilbert spaces are by far the nicest. If you use or study orthonormal basis expansion (Section 15.8) then you will start to see why this is true.

15.7 Types of Basis¹²

15.7.1 Normalized Basis

Definition 24: Normalized Basis

a basis (Section 5.1.3: Basis) $\{b_i\}$ where each b_i has unit norm

$$\|b_i\| = 1, \quad i \in \mathbb{Z} \quad (15.19)$$

NOTE: The concept of basis applies to all vector spaces (Section 15.1). The concept of **normalized basis** applies only to normed spaces (Section 15.2).

You can always normalize a basis: just multiply each basis vector by a constant, such as $\frac{1}{\|b_i\|}$

Example 15.6

We are given the following basis:

$$\{b_0, b_1\} = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

Normalized with ℓ^2 norm:

$$\begin{aligned} \tilde{b}_0 &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ \tilde{b}_1 &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \end{aligned}$$

Normalized with ℓ^1 norm:

$$\begin{aligned} \tilde{b}_0 &= \frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ \tilde{b}_1 &= \frac{1}{2} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \end{aligned}$$

15.7.2 Orthogonal Basis

Definition 25: Orthogonal Basis

a basis $\{b_i\}$ in which the elements are **mutually orthogonal**

$$\langle b_i, b_j \rangle = 0, \quad i \neq j$$

¹²This content is available online at <http://cnx.org/content/m10772/2.5/>.

NOTE: The concept of orthogonal basis applies only to Hilbert Spaces.

Example 15.7

Standard basis for \mathbb{R}^2 , also referred to as $\ell^2([0, 1])$:

$$b_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$b_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\langle b_0, b_1 \rangle = \sum_{i=0}^1 (b_0[i] b_1[i]) = 1 \times 0 + 0 \times 1 = 0$$

Example 15.8

Now we have the following basis and relationship:

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\} = \{h_0, h_1\}$$

$$\langle h_0, h_1 \rangle = 1 \times 1 + 1 \times (-1) = 0$$

15.7.3 Orthonormal Basis

Pulling the previous two sections (definitions) together, we arrive at the most important and useful basis type:

Definition 26: Orthonormal Basis

a basis that is both **normalized** and **orthogonal**

$$\| b_i \| = 1 \quad , \quad i \in \mathbb{Z}$$

$$\langle b_i, b_j \rangle = 0 \quad , \quad i \neq j$$

NOTATION: We can shorten these two statements into one:

$$\langle b_i, b_j \rangle = \delta_{ij}$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Where δ_{ij} is referred to as the Kronecker delta function (Section 1.5) and is also often written as $\delta[i - j]$.

Example 15.9: Orthonormal Basis Example #1

$$\{b_0, b_2\} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

Example 15.10: Orthonormal Basis Example #2

$$\{b_0, b_2\} = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

Example 15.11: Orthonormal Basis Example #3

$$\{b_0, b_2\} = \left\{ \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

15.7.3.1 Beauty of Orthonormal Bases

Orthonormal bases are very easy to deal with! If $\{b_i\}$ is an orthonormal basis, we can write for any x

$$x = \sum_i (\alpha_i b_i) \tag{15.20}$$

It is easy to find the α_i :

$$\begin{aligned} \langle x, b_i \rangle &= \langle \sum_k (\alpha_k b_k), b_i \rangle \\ &= \sum_k (\alpha_k \langle b_k, b_i \rangle) \end{aligned} \tag{15.21}$$

where in the above equation we can use our knowledge of the delta function to reduce this equation:

$$\begin{aligned} \langle b_k, b_i \rangle &= \delta_{ik} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases} \\ \langle x, b_i \rangle &= \alpha_i \end{aligned} \tag{15.22}$$

Therefore, we can conclude the following important equation for x :

$$x = \sum_i (\langle x, b_i \rangle b_i) \tag{15.23}$$

The α_i 's are easy to compute (no interaction between the b_i 's)

Example 15.12

Given the following basis:

$$\{b_0, b_1\} = \left\{ \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

represent $x = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$

Example 15.13: Slightly Modified Fourier Series

We are given the basis

$$\left\{ \frac{1}{\sqrt{T}} e^{j\omega_0 n t} \right\}_{n=-\infty}^{\infty}$$

on $L^2([0, T])$ where $T = \frac{2\pi}{\omega_0}$.

$$f(t) = \sum_{n=-\infty}^{\infty} \left(\langle f, e^{j\omega_0 n t} \rangle e^{j\omega_0 n t} \frac{1}{\sqrt{T}} \right)$$

Where we can calculate the above inner product in L^2 as

$$\langle f, e^{j\omega_0 n t} \rangle = \frac{1}{\sqrt{T}} \int_0^T f(t) e^{j\omega_0 n t} dt = \frac{1}{\sqrt{T}} \int_0^T f(t) e^{-j\omega_0 n t} dt$$

15.7.3.2 Orthonormal Basis Expansions in a Hilbert Space

Let $\{b_i\}$ be an orthonormal basis for a Hilbert space H . Then, for any $x \in H$ we can write

$$x = \sum_i (\alpha_i b_i) \tag{15.24}$$

where $\alpha_i = \langle x, b_i \rangle$.

- "Analysis": decomposing x in term of the b_i

$$\alpha_i = \langle x, b_i \rangle \tag{15.25}$$

- "Synthesis": building x up out of a weighted combination of the b_i

$$x = \sum_i (\alpha_i b_i) \tag{15.26}$$

This is an unsupported media type. To view, please see <http://cnx.org/content/m10772/latest/ONB.llb>

15.8 Orthonormal Basis Expansions¹³**15.8.1 Main Idea**

When working with signals many times it is helpful to break up a signal into smaller, more manageable parts. Hopefully by now you have been exposed to the concept of eigenvectors (Section 5.2) and there use in decomposing a signal into one of its possible basis. By doing this we are able to simplify our calculations of signals and systems through eigenfunctions of LTI systems (Section 5.5).

Now we would like to look at an alternative way to represent signals, through the use of **orthonormal basis**. We can think of orthonormal basis as a set of building blocks we use to construct functions. We will build up the signal/vector as a weighted sum of basis elements.

¹³This content is available online at <http://cnx.org/content/m10760/2.4/>.

Example 15.14

The complex sinusoids $\frac{1}{\sqrt{T}}e^{j\omega_0 nt}$ for all $-\infty < n < \infty$ form an orthonormal basis for $L^2([0, T])$.

In our Fourier series (Section 6.2) equation, $f(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j\omega_0 nt})$, the $\{c_n\}$ are just another representation of $f(t)$.

NOTE: For signals/vectors in a Hilbert Space, the expansion coefficients are easy to find.

15.8.2 Alternate Representation

Recall our definition of a **basis**: A set of vectors $\{b_i\}$ in a vector space S is a basis if

1. The b_i are linearly independent.
2. The b_i span (Section 5.1.2: Span) S . That is, we can find $\{\alpha_i\}$, where $\alpha_i \in \mathbb{C}$ (scalars) such that

$$\mathbf{x} = \sum_i (\alpha_i b_i) \quad , \quad x \in S \quad (15.27)$$

where \mathbf{x} is a vector in S , α is a scalar in \mathbb{C} , and \mathbf{b} is a vector in S .

Condition 2 in the above definition says we can **decompose** any vector in terms of the $\{b_i\}$. Condition 1 ensures that the decomposition is **unique** (think about this at home).

NOTE: The $\{\alpha_i\}$ provide an alternate representation of \mathbf{x} .

Example 15.15

Let us look at simple example in \mathbb{R}^2 , where we have the following vector:

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\text{Standard Basis: } \{e_0, e_1\} = \{(1, 0)^T, (0, 1)^T\}$$

$$\mathbf{x} = e_0 + 2e_1$$

$$\text{Alternate Basis: } \{h_0, h_1\} = \{(1, 1)^T, (1, -1)^T\}$$

$$\mathbf{x} = \frac{3}{2}h_0 + \frac{-1}{2}h_1$$

In general, given a basis $\{b_0, b_1\}$ and a vector $\mathbf{x} \in \mathbb{R}^2$, how do we find the α_0 and α_1 such that

$$\mathbf{x} = \alpha_0 b_0 + \alpha_1 b_1 \quad (15.28)$$

15.8.3 Finding the Alphas

Now let us address the question posed above about finding α_i 's in general for \mathbb{R}^2 . We start by rewriting (15.28) so that we can stack our b_i 's as columns in a 2×2 matrix.

$$\begin{pmatrix} \mathbf{x} \end{pmatrix} = \alpha_0 \begin{pmatrix} b_0 \end{pmatrix} + \alpha_1 \begin{pmatrix} b_1 \end{pmatrix} \quad (15.29)$$

$$\begin{pmatrix} \mathbf{x} \end{pmatrix} = \begin{pmatrix} \vdots & \vdots \\ b_0 & b_1 \\ \vdots & \vdots \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \quad (15.30)$$

Example 15.16

Here is a simple example, which shows a little more detail about the above equations.

$$\begin{aligned} \begin{pmatrix} x[0] \\ x[1] \end{pmatrix} &= \alpha_0 \begin{pmatrix} b_0[0] \\ b_0[1] \end{pmatrix} + \alpha_1 \begin{pmatrix} b_1[0] \\ b_1[1] \end{pmatrix} \\ &= \begin{pmatrix} \alpha_0 b_0[0] + \alpha_1 b_1[0] \\ \alpha_0 b_0[1] + \alpha_1 b_1[1] \end{pmatrix} \end{aligned} \quad (15.31)$$

$$\begin{pmatrix} x[0] \\ x[1] \end{pmatrix} = \begin{pmatrix} b_0[0] & b_1[0] \\ b_0[1] & b_1[1] \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \quad (15.32)$$

15.8.3.1 Simplifying our Equation

To make notation simpler, we define the following two items from the above equations:

- **Basis Matrix:**

$$B = \begin{pmatrix} \vdots & \vdots \\ b_0 & b_1 \\ \vdots & \vdots \end{pmatrix}$$

- **Coefficient Vector:**

$$\alpha = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$$

This gives us the following, concise equation:

$$\mathbf{x} = B\alpha \quad (15.33)$$

which is equivalent to $\mathbf{x} = \sum_{i=0}^1 (\alpha_i b_i)$.

Example 15.17

Given a standard basis, $\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$, then we have the following basis matrix:

$$B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

To get the α_i 's, we solve for the coefficient vector in (15.33)

$$\alpha = B^{-1}\mathbf{x} \quad (15.34)$$

Where B^{-1} is the inverse matrix¹⁴ of B .

15.8.3.2 Examples

Example 15.18

Let us look at the standard basis first and try to calculate α from it.

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

Where I is the **identity matrix**. In order to solve for α let us find the inverse of B first (which is obviously very trivial in this case):

$$B^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Therefore we get,

$$\alpha = B^{-1}\mathbf{x} = \mathbf{x}$$

Example 15.19

Let us look at a ever-so-slightly more complicated basis of $\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\} = \{h_0, h_1\}$ Then our basis matrix and inverse basis matrix becomes:

$$B = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$B^{-1} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

and for this example it is given that

$$\mathbf{x} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Now we solve for α

$$\alpha = B^{-1}\mathbf{x} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 2.5 \\ 0.5 \end{pmatrix}$$

and we get

$$\mathbf{x} = 2.5h_0 + 0.5h_1$$

Exercise 15.1

Now we are given the following basis matrix and \mathbf{x} :

$$\{b_0, b_1\} = \left\{ \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix} \right\}$$

(Solution on p. 301.)

¹⁴"Matrix Inversion" <<http://cnx.org/content/m2113/latest/>>

$$\mathbf{x} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

For this problem, make a sketch of the bases and then represent \mathbf{x} in terms of b_0 and b_1 .

NOTE: A change of basis simply looks at \mathbf{x} from a "different perspective." B^{-1} **transforms** \mathbf{x} from the standard basis to our new basis, $\{b_0, b_1\}$. Notice that this is a totally mechanical procedure.

15.8.4 Extending the Dimension and Space

We can also extend all these ideas past just \mathbb{R}^2 and look at them in \mathbb{R}^n and \mathbb{C}^n . This procedure extends naturally to higher (> 2) dimensions. Given a basis $\{b_0, b_1, \dots, b_{n-1}\}$ for \mathbb{R}^n , we want to find $\{\alpha_0, \alpha_1, \dots, \alpha_{n-1}\}$ such that

$$\mathbf{x} = \alpha_0 b_0 + \alpha_1 b_1 + \dots + \alpha_{n-1} b_{n-1} \quad (15.35)$$

Again, we will set up a basis matrix

$$B = \begin{pmatrix} b_0 & b_1 & b_2 & \dots & b_{n-1} \end{pmatrix}$$

where the columns equal the basis vectors and it will always be an $n \times n$ matrix (although the above matrix does not appear to be square since we left terms in vector notation). We can then proceed to rewrite (15.33)

$$\mathbf{x} = \begin{pmatrix} b_0 & b_1 & \dots & b_{n-1} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{n-1} \end{pmatrix} = B\alpha$$

and

$$\alpha = B^{-1}\mathbf{x}$$

15.9 Function Space¹⁵

We can also find basis vectors (Section 15.8) for vector spaces (Section 15.1) other than \mathbb{R}^n .

Let P_n be the vector space of n -th order polynomials on $(-1, 1)$ with real coefficients (verify P_2 is a v.s. at home).

Example 15.20

$P_2 = \{\text{all quadratic polynomials}\}$. Let $b_0(t) = 1$, $b_1(t) = t$, $b_2(t) = t^2$.
 $\{b_0(t), b_1(t), b_2(t)\}$ **span** P_2 , i.e. you can write any $f(t) \in P_2$ as

$$f(t) = \alpha_0 b_0(t) + \alpha_1 b_1(t) + \alpha_2 b_2(t)$$

for some $\alpha_i \in \mathbb{R}$.

NOTE: P_2 is 3 dimensional.

¹⁵This content is available online at <http://cnx.org/content/m10770/2.4/>.

$$f(t) = t^2 - 3t - 4$$

Alternate basis

$$\{b_0(t), b_1(t), b_2(t)\} = \left\{1, t, \frac{1}{2}(3t^2 - 1)\right\}$$

write $f(t)$ in terms of this new basis $d_0(t) = b_0(t)$, $d_1(t) = b_1(t)$, $d_2(t) = \frac{3}{2}b_2(t) - \frac{1}{2}b_0(t)$.

$$f(t) = t^2 - 3t - 4 = 4b_0(t) - 3b_1(t) + b_2(t)$$

$$f(t) = \beta_0 d_0(t) + \beta_1 d_1(t) + \beta_2 d_2(t) = \beta_0 b_0(t) + \beta_1 b_1(t) + \beta_2 \left(\frac{3}{2}b_2(t) - \frac{1}{2}b_0(t)\right)$$

$$f(t) = \left(\beta_0 - \frac{1}{2}\right) b_0(t) + \beta_1 b_1(t) + \frac{3}{2}\beta_2 b_2(t)$$

so

$$\beta_0 - \frac{1}{2} = 4$$

$$\beta_1 = -3$$

$$\frac{3}{2}\beta_2 = 1$$

then we get

$$f(t) = 4.5d_0(t) - 3d_1(t) + \frac{2}{3}d_2(t)$$

Example 15.21

$e^{j\omega_0 nt}|_{n=-\infty}^{\infty}$ is a basis for $L^2([0, T])$, $T = \frac{2\pi}{\omega_0}$, $f(t) = \sum_n (C_n e^{j\omega_0 nt})$.

We calculate the expansion coefficients with

"change of basis" formula

$$C_n = \frac{1}{T} \int_0^T f(t) e^{-j\omega_0 nt} dt \quad (15.36)$$

NOTE: There are an infinite number of elements in the basis set, that means $L^2([0, T])$ is infinite dimensional (scary!).

Infinite-dimensional spaces are hard to visualize. We can get a handle on the intuition by recognizing they share many of the same mathematical properties with finite dimensional spaces. Many concepts apply to both (like "basis expansion"). Some don't (change of basis isn't a nice matrix formula).

15.10 Haar Wavelet Basis¹⁶

15.10.1 Introduction

Fourier series (Section 6.2) is a useful orthonormal representation (Section 15.8) on $L^2([0, T])$ especially for inputs into LTI systems. However, it is ill suited for some applications, i.e. image processing (recall Gibb's phenomena (Section 6.11)).

Wavelets, discovered in the last 15 years, are another kind of basis for $L^2([0, T])$ and have many nice properties.

¹⁶This content is available online at <<http://cnx.org/content/m10764/2.7/>>.

15.10.2 Basis Comparisons

Fourier series - c_n give frequency information. Basis functions last the entire interval.

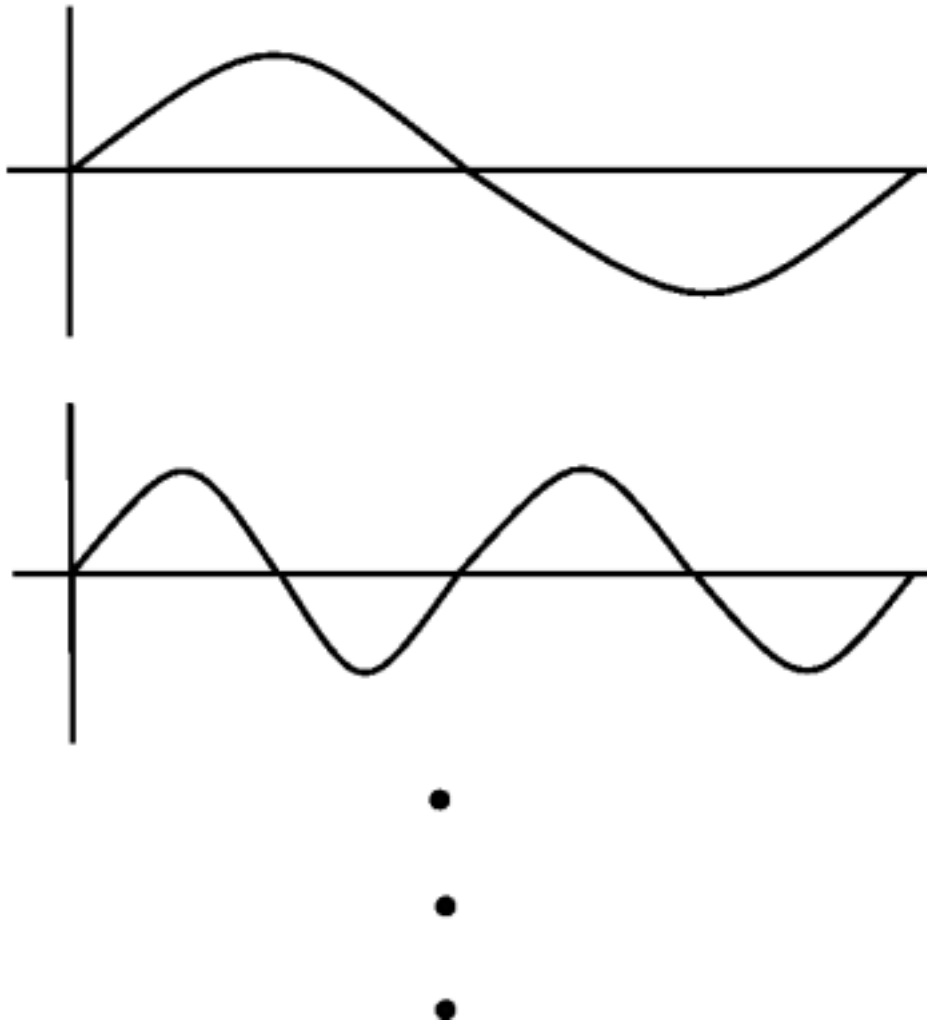


Figure 15.18: Fourier basis functions

Wavelets - basis functions give frequency info but are *local* in time.

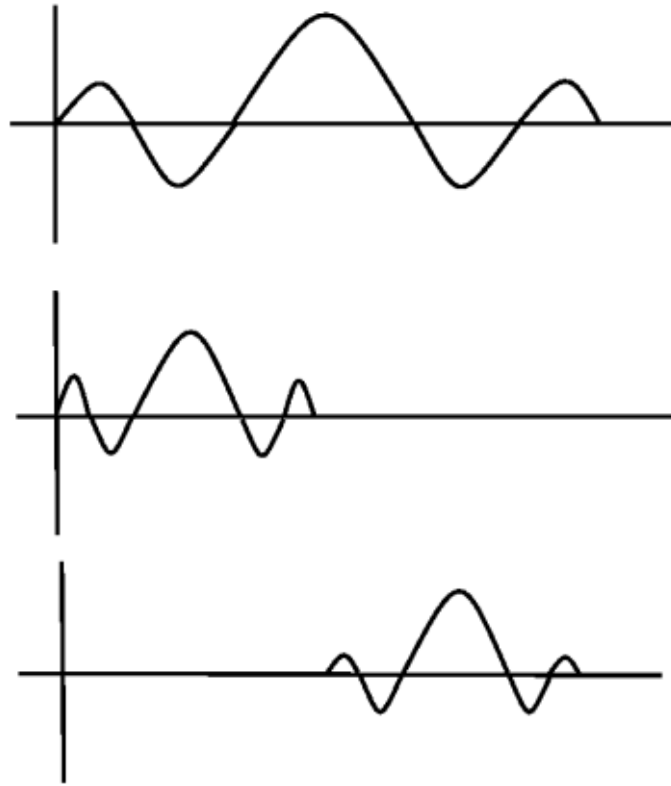


Figure 15.19: Wavelet basis functions

In Fourier basis, the basis functions are *harmonic multiples* of $e^{j\omega_0 t}$

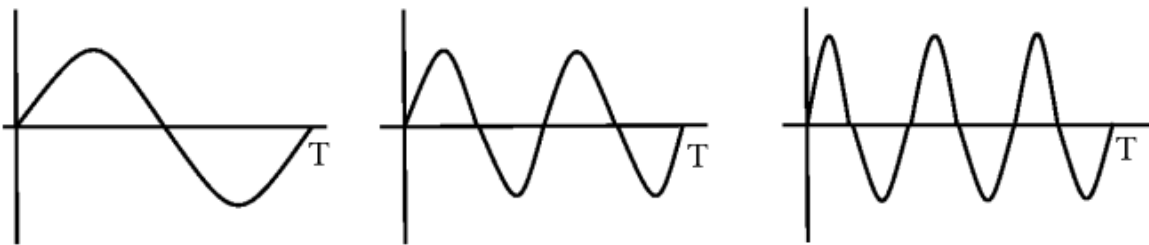


Figure 15.20: $basis = \left\{ \frac{1}{\sqrt{T}} e^{j\omega_0 n t} \right\}$

In Haar wavelet basis¹⁷, the basis functions are *scaled and translated* versions of a "mother wavelet" $\psi(t)$.

¹⁷"The Haar System as an Example of DWT" <<http://cnx.org/content/m10437/latest/>>

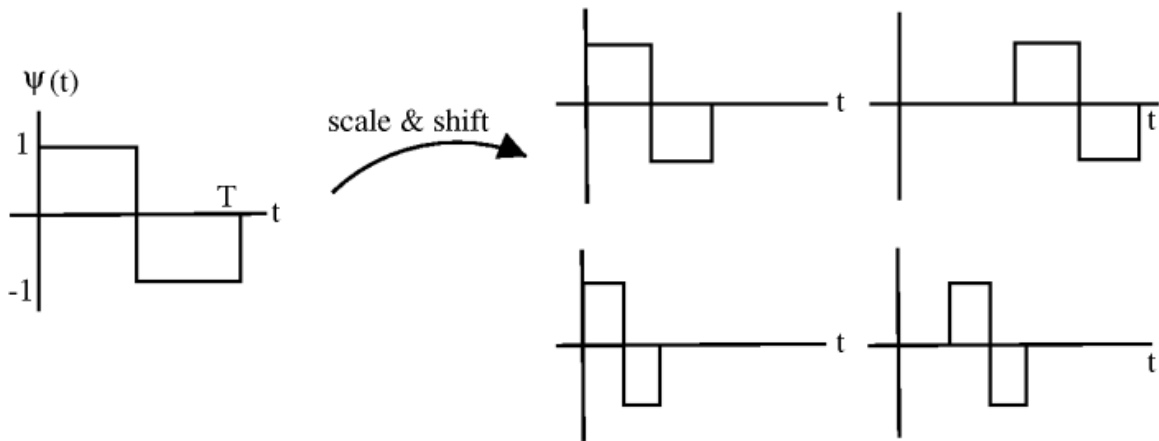


Figure 15.21

Basis functions $\{\psi_{j,k}(t)\}$ are indexed by a *scale* j and a *shift* k .

Let $\phi(t) = 1$, $0 \leq t < T$. Then $\left\{ \phi(t), 2^{\frac{j}{2}} \psi(2^j t - k) \mid j \in \mathbb{Z} \text{ and } k = 0, 1, 2, \dots, 2^j - 1 \right\}$

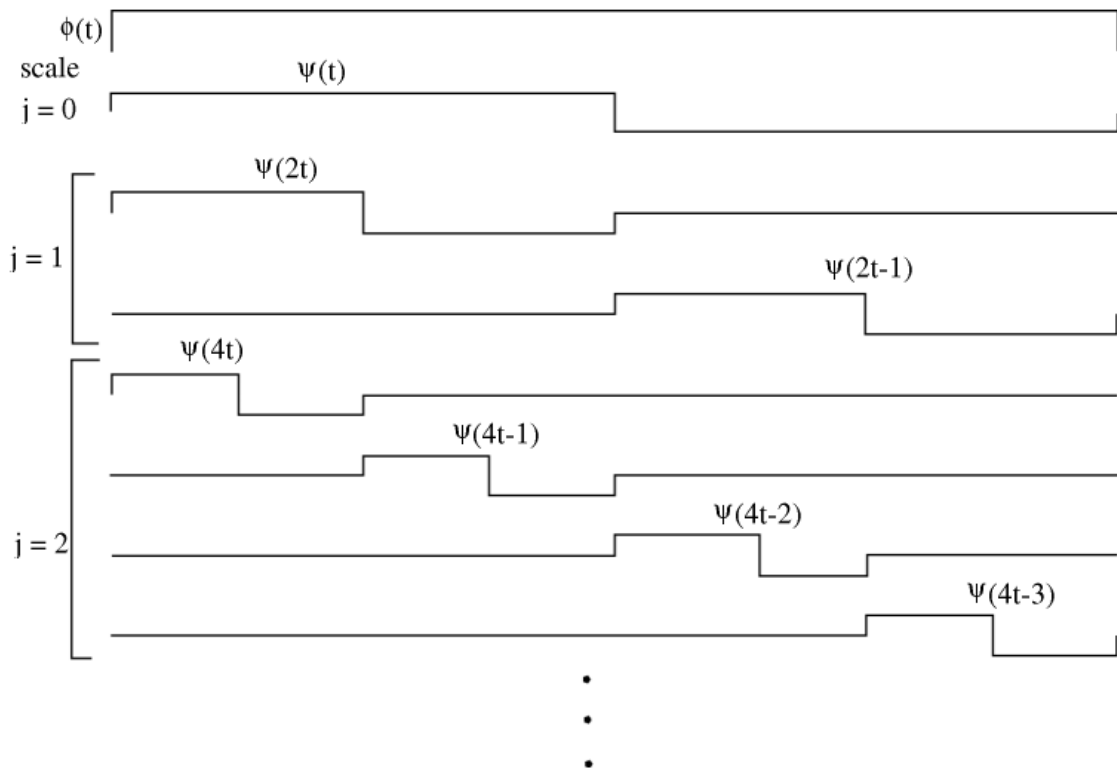


Figure 15.22

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < \frac{T}{2} \\ -1 & \text{if } \frac{T}{2} \leq t < T \end{cases} \tag{15.37}$$

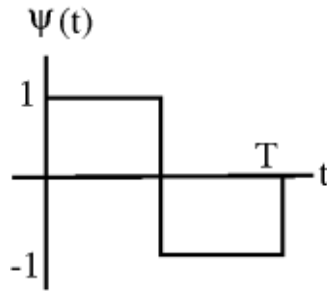


Figure 15.23

Let $\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k)$

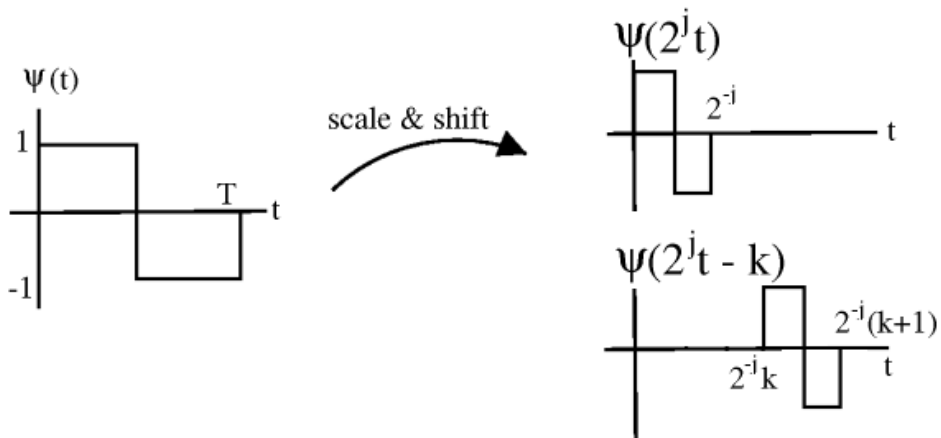


Figure 15.24

Larger $j \rightarrow$ "skinnier" basis function, $j = \{0, 1, 2, \dots\}$, 2^j shifts at each scale: $k = 0, 1, \dots, 2^j - 1$
 Check: each $\psi_{j,k}(t)$ has unit energy

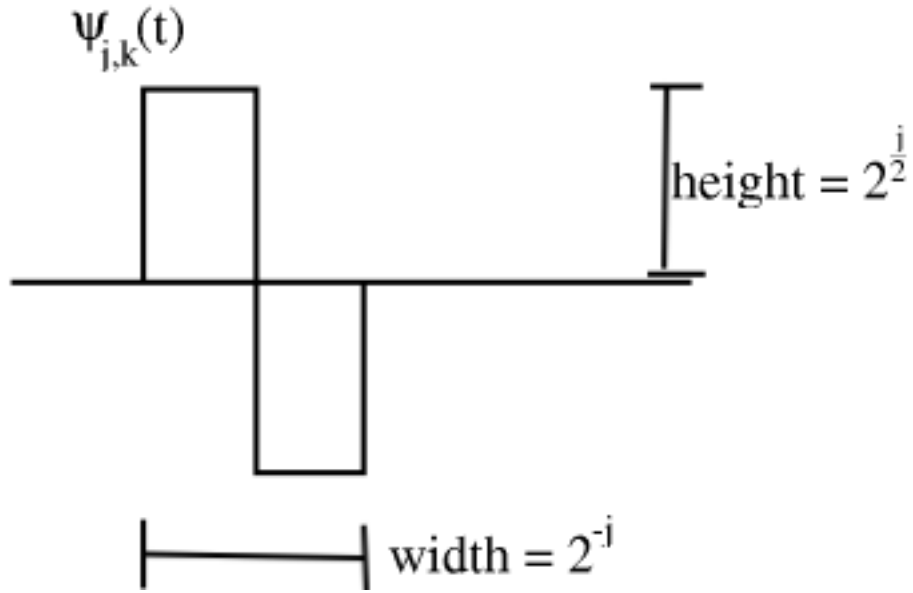


Figure 15.25

$$\int \psi_{j,k}^2(t) dt = 1 \Rightarrow \|\psi_{j,k}(t)\|_2 = 1 \quad (15.38)$$

Any two basis functions are orthogonal.

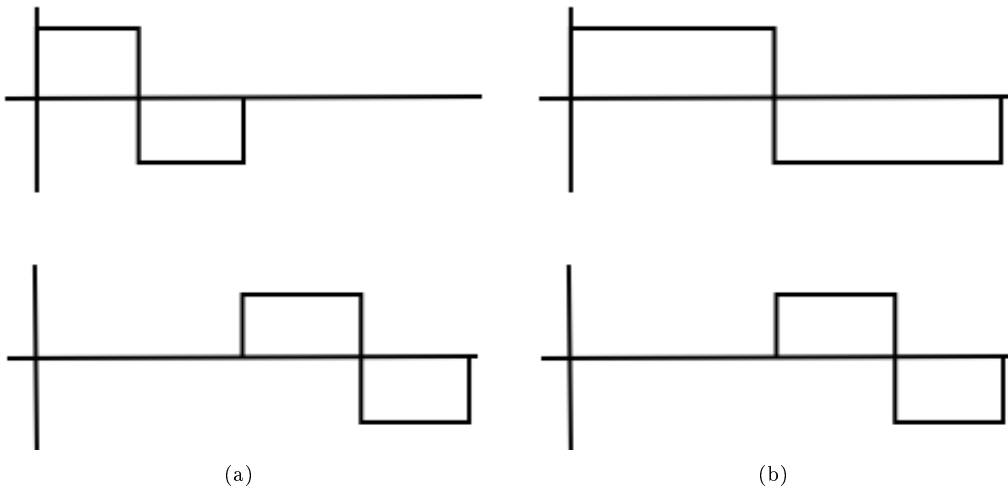


Figure 15.26: Integral of product = 0 (a) Same scale (b) Different scale

Also, $\{\psi_{j,k}, \phi\}$ span $L^2([0, T])$

15.10.3 Haar Wavelet Transform

Using what we know about Hilbert spaces (Section 15.3): For any $f(t) \in L^2([0, T])$, we can write

Synthesis

$$f(t) = \sum_j \left(\sum_k (w_{j,k} \psi_{j,k}(t)) \right) + c_0 \phi(t) \quad (15.39)$$

Analysis

$$w_{j,k} = \int_0^T f(t) \psi_{j,k}(t) dt \quad (15.40)$$

$$c_0 = \int_0^T f(t) \phi(t) dt \quad (15.41)$$

NOTE: the $w_{j,k}$ are real

The Haar transform is *super useful* especially in *image compression*

Example 15.22

This demonstration lets you create a signal by combining Haar basis functions, illustrating the synthesis equation of the Haar Wavelet Transform. See here¹⁸ for instructions on how to use the demo.

This is an unsupported media type. To view, please see <http://cnx.org/content/m10764/latest/HaarSyn.llb>

15.11 Orthonormal Bases in Real and Complex Spaces¹⁹

15.11.1 Notation

Transpose operator A^T flips the matrix across its diagonal.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

$$A^T = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix}$$

Column i of A is row i of A^T

¹⁸How to use the LabVIEW demos" <<http://cnx.org/content/m11550/latest/>>

¹⁹This content is available online at <<http://cnx.org/content/m10765/2.7/>>.

Recall, inner product²⁰

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

$$\mathbf{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

$$\mathbf{x}^T \mathbf{y} = \begin{pmatrix} x_0 & x_1 & \dots & x_{n-1} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \sum (\mathbf{x}_i \mathbf{y}_i) = \langle \mathbf{y}, \mathbf{x} \rangle$$

on \mathbb{R}^n

Hermitian transpose A^H , transpose and conjugate

$$A^H = A^{T*}$$

$$\langle \mathbf{y}, \mathbf{x} \rangle = \mathbf{x}^H \mathbf{y} = \sum (\mathbf{x}_i \mathbf{y}_i^*)$$

on \mathbb{C}^n

Now, let $\{b_0, b_1, \dots, b_{n-1}\}$ be an orthonormal basis (Section 15.7.3: Orthonormal Basis) for \mathbb{C}^n

$$i = \{0, 1, \dots, n-1\} \quad \langle b_i, b_i \rangle = 1 \quad ,$$

$$i \neq j \quad \langle b_i, b_j \rangle = b_j^H b_i = 0$$

Basis matrix:

$$B = \begin{pmatrix} \vdots & \vdots & & \vdots \\ b_0 & b_1 & \dots & b_{n-1} \\ \vdots & \vdots & & \vdots \end{pmatrix}$$

Now,

$$B^H B = \begin{pmatrix} \dots & b_0^H & \dots \\ \dots & b_1^H & \dots \\ & \vdots & \\ \dots & b_{n-1}^H & \dots \end{pmatrix} \begin{pmatrix} \vdots & \vdots & & \vdots \\ b_0 & b_1 & \dots & b_{n-1} \\ \vdots & \vdots & & \vdots \end{pmatrix} = \begin{pmatrix} b_0^H b_0 & b_0^H b_1 & \dots & b_0^H b_{n-1} \\ b_1^H b_0 & b_1^H b_1 & \dots & b_1^H b_{n-1} \\ \vdots & \vdots & & \vdots \\ b_{n-1}^H b_0 & b_{n-1}^H b_1 & \dots & b_{n-1}^H b_{n-1} \end{pmatrix}$$

²⁰"Conclusion" <<http://cnx.org/content/m10775/latest/>>

For orthonormal basis with basis matrix B

$$B^H = B^{-1}$$

($B^T = B^{-1}$ in \mathbb{R}^n) B^H is easy to calculate while B^{-1} is hard to calculate.

So, to find $\{\alpha_0, \alpha_1, \dots, \alpha_{n-1}\}$ such that

$$\mathbf{x} = \sum (\alpha_i b_i)$$

Calculate

$$\alpha = B^{-1} \mathbf{x} \Rightarrow \alpha = B^H \mathbf{x}$$

Using an orthonormal basis we rid ourselves of the inverse operation.

15.12 Plancharel and Parseval's Theorems²¹

15.12.1 Plancharel Theorem

Theorem 15.2: Plancharel Theorem

The inner product of two vectors/signals is the same as the ℓ^2 inner product of their expansion coefficients.

Let $\{b_i\}$ be an orthonormal basis for a Hilbert Space H . $x \in H, y \in H$

$$x = \sum (\alpha_i b_i)$$

$$y = \sum (\beta_i b_i)$$

then

$$\langle x, y \rangle_H = \sum (\alpha_i \beta_i^*)$$

Example

Applying the Fourier Series, we can go from $f(t)$ to $\{c_n\}$ and $g(t)$ to $\{d_n\}$

$$\int_0^T f(t) g(t)^* dt = \sum_{n=-\infty}^{\infty} (c_n d_n^*)$$

inner product in time-domain = inner product of Fourier coefficients.

Proof:

$$x = \sum (\alpha_i b_i)$$

$$y = \sum (\beta_j b_j)$$

$$\langle x, y \rangle_H = \langle \sum (\alpha_i b_i), \sum (\beta_j b_j) \rangle = \left(\sum \alpha_i \right) \langle b_i, \sum (\beta_j b_j) \rangle = \left(\sum \alpha_i \right) \left(\sum (\beta_j^*) \right) \langle b_i, b_j \rangle = \sum (\alpha_i \beta_i^*)$$

by using inner product rules (p. 269)

NOTE: $\langle b_i, b_j \rangle = 0$ when $i \neq j$ and $\langle b_i, b_j \rangle = 1$ when $i = j$

If Hilbert space H has a ONB, then inner products are equivalent to inner products in ℓ^2 .

All H with ONB are somehow equivalent to ℓ^2 .

POINT OF INTEREST: square-summable sequences are important

²¹This content is available online at <http://cnx.org/content/m10769/2.5/>.

15.12.2 Parseval's Theorem

Theorem 15.3: Parseval's Theorem

Energy of a signal = sum of squares of it's expansion coefficients

Let $x \in H$, $\{b_i\}$ ONB

$$x = \sum (\alpha_i b_i)$$

Then

$$(\| x \|_H)^2 = \sum (|\alpha_i|^2)$$

Proof: Directly from Plancharel

$$(\| x \|_H)^2 = \langle x, x \rangle_H = \sum (\alpha_i \alpha_i^*) = \sum (|\alpha_i|^2)$$

Example 15.24

Fourier Series $\frac{1}{\sqrt{T}} e^{jw_0 t}$

$$f(t) = \frac{1}{\sqrt{T}} \sum \left(c_n \frac{1}{\sqrt{T}} e^{jw_0 n t} \right)$$

$$\int_0^T (|f(t)|)^2 dt = \sum_{n=-\infty}^{\infty} (|c_n|^2)$$

This is an unsupported media type. To view, please see <http://cnx.org/content/m10769/latest/ParsevalsTheorem.llb>

15.13 Approximation and Projections in Hilbert Space²²

15.13.1 Introduction

Given a line 'l' and a point 'p' in the plane, what's the closest point 'm' to 'p' on 'l'?

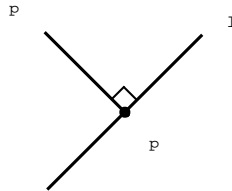


Figure 15.27: Figure of point 'p' and line 'l' mentioned above.

Same problem: Let x and v be vectors in \mathbb{R}^2 . Say $\| v \| = 1$. For what value of α is $\| x - \alpha v \|_2$ minimized? (what point in $\text{span}\{v\}$ best approximates x ?)

²²This content is available online at <http://cnx.org/content/m10766/2.7/>.

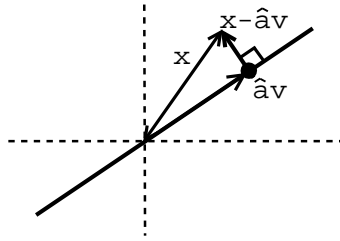


Figure 15.28

The condition is that $x - \hat{\alpha}v$ and αv are **orthogonal**.

15.13.2 Calculating α

How to calculate $\hat{\alpha}$?

We know that $(x - \hat{\alpha}v)$ is perpendicular to every vector in $\text{span}\{v\}$, so

$$\langle x - \hat{\alpha}v, \beta v \rangle = 0, \quad \forall \beta$$

$$\beta^* \langle x, v \rangle - \hat{\alpha} \beta^* \langle v, v \rangle = 0$$

because $\langle v, v \rangle = 1$, so

$$\langle x, v \rangle - \hat{\alpha} = 0 \Rightarrow \hat{\alpha} = \langle x, v \rangle$$

Closest vector in $\text{span}\{v\} = \langle x, v \rangle v$, where $\langle x, v \rangle v$ is the projection of \mathbf{x} onto \mathbf{v} .

We can do the same thing in higher dimensions.

Exercise 15.2

(Solution on p. 301.)

Let $V \subset H$ be a subspace of a Hilbert space (Section 15.3) H . Let $x \in H$ be given. Find the $y \in V$ that **best approximates** x . i.e., $\|x - y\|$ is minimized.

Example 15.25

$$x \in \mathbb{R}^3, V = \text{span} \left(\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\} \right), x = \begin{pmatrix} a \\ b \\ c \end{pmatrix}. \text{ So,}$$

$$y = \sum_{i=1}^2 (\langle x, b_i \rangle b_i) = a \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ b \\ 0 \end{pmatrix}$$

Example 15.26

$V = \{\text{space of periodic signals with frequency no greater than } 3\omega_0\}$. Given periodic $f(t)$, what is the signal in V that best approximates f ?

1. $\{ \frac{1}{\sqrt{T}} e^{j\omega_0 kt}, k = -3, -2, \dots, 2, 3 \}$ is an ONB for V
2. $g(t) = \frac{1}{T} \sum_{k=-3}^3 (\langle f(t), e^{j\omega_0 kt} \rangle e^{j\omega_0 kt})$ is the closest signal in V to $f(t) \Rightarrow$ reconstruct $f(t)$ using only 7 terms of its Fourier series (Section 6.2).

Example 15.27

Let $V = \{\text{functions piecewise constant between the integers}\}$

1. ONB for V .

$$b_i = \begin{cases} 1 & \text{if } i-1 \leq t < i \\ 0 & \text{otherwise} \end{cases}$$

where $\{b_i\}$ is an ONB.

Best piecewise constant approximation?

$$g(t) = \sum_{i=-\infty}^{\infty} (\langle f, b_i \rangle b_i)$$

$$\langle f, b_i \rangle = \int_{-\infty}^{\infty} f(t) b_i(t) dt = \int_{i-1}^i f(t) dt$$

Example 15.28

This demonstration explores approximation using a Fourier basis and a Haar Wavelet basis. See here²³ for instructions on how to use the demo.

This is an unsupported media type. To view, please see
<http://cnx.org/content/m10766/latest/Approximation.llb>

²³"How to use the LabVIEW demos" <<http://cnx.org/content/m11550/latest/>>

Solutions to Exercises in Chapter 15

Solution to Exercise 15.1 (p. 286)

In order to represent \mathbf{x} in terms of b_0 and b_1 we will follow the same steps we used in the above example.

$$B = \begin{pmatrix} 1 & 2 \\ 3 & 0 \end{pmatrix}$$

$$B^{-1} = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{-1}{6} \end{pmatrix}$$

$$\alpha = B^{-1}\mathbf{x} = \begin{pmatrix} 1 \\ \frac{2}{3} \end{pmatrix}$$

And now we can write \mathbf{x} in terms of b_0 and b_1 .

$$\mathbf{x} = b_0 + \frac{2}{3}b_1$$

And we can easily substitute in our known values of b_0 and b_1 to verify our results.

Solution to Exercise 15.2 (p. 299)

1. Find an orthonormal basis (Section 15.7.3: Orthonormal Basis) $\{b_1, \dots, b_k\}$ for V
2. Project x onto V using

$$y = \sum_{i=1}^k (\langle x, b_i \rangle b_i)$$

then y is the closest point in V to x and $(x-y) \perp V$ ($\langle x-y, v \rangle = 0$, $\forall v \in V$)

Chapter 16

Homework Sets

16.1 Homework #1¹

DUE DATE: Noon, Thursday, September 5, 2002

16.1.1 Assignment 1

Homework, tests, and solutions from previous offerings of this course are off limits, under the honor code.

16.1.1.1 Problem 1

Form a study group of 3-4 members. With your group, discuss and synthesize the major *themes* of this week of lectures. Turn in a one page summary of your discussion. You need turn in only one summary per group, but include the names of all group members. Please do not write up just a "table of contents."

16.1.1.2 Problem 2

Construct a WWW page (with your *picture*) and email Mike Wakin (wakin@rice.edu) your name (as you want it to appear on the class web page) and the URL. If you need assistance setting up your page or taking/scanning a picture (both are easy!), ask your classmates.

16.1.1.3 Problem 3: Learning Styles

Follow this learning styles link² (also found on the Elec 301 web page³) and learn about the basics of learning styles. Write a short summary of what you learned. Also, complete the "Index of learning styles" self-scoring test on the web and bring your results to class.

16.1.1.4 Problem 4

Make sure you know the material in *Lathi*, Chapter B, Sections 1-4, 6.1, 6.2, 7. Specifically, be sure to review topics such as:

- complex arithmetic (adding, multiplying, powers)
- finding (complex) roots of polynomials
- complex plane⁴ and plotting roots

¹This content is available online at <<http://cnx.org/content/m10826/2.9/>>.

²http://www2.ncsu.edu/unity/lockers/users/f/felder/public/Learning_Styles.html

³<http://www-dsp.rice.edu/courses/elec301/>

⁴"The Complex Plane" <<http://cnx.org/content/m10596/latest/>>

- vectors (adding, inner products)

16.1.1.5 Problem 5: Complex Number Applet

Reacquaint yourself with complex numbers⁵ by going to the course applets web page⁶ and clicking on the Complex Numbers applet (may take a few seconds to load).

(a) Change the default add function to exponential (exp). Click on the complex plane to get a blue arrow, which is your complex number z . Click again anywhere on the complex plane to get a yellow arrow, which is equal to e^z . Now drag the tip of the blue arrow along the unit circle on with $|z| = 1$ (smaller circle). For which values of z on the unit circle does e^z also lie on the unit circle? Why?

(b) Experiment with the functions absolute (abs), real part (re), and imaginary part (im) and report your findings.

16.1.1.6 Problem 6: Complex Arithmetic

Reduce the following to the **Cartesian** form, $a + jb$. Do *not* use your calculator!

- $\left(\frac{-1-j}{\sqrt{2}}\right)^{20}$
- $\frac{1+2j}{3+4j}$
- $\frac{1+\sqrt{3}j}{\sqrt{3}-j}$
- \sqrt{j}
- j^j

16.1.1.7 Problem 7: Roots of Polynomials

Find the roots of each of the following polynomials (show your work). Use MATLAB to check your answer with the `roots` command and to plot the roots in the complex plane. Mark the root locations with an 'o'. Put all of the roots on the same plot and identify the corresponding polynomial ($a, b, etc...$).

- $z^2 - 4z$
- $z^2 - 4z + 4$
- $z^2 - 4z + 8$
- $z^2 + 8$
- $z^2 + 4z + 8$
- $2z^2 + 4z + 8$

16.1.1.8 Problem 8: Nth Roots of Unity

$e^{\frac{j2\pi}{N}}$ is called an **Nth Root of Unity**.

- Why?
- Let $z = e^{\frac{j2\pi}{7}}$. Draw $\{z, z^2, \dots, z^7\}$ in the complex plane.
- Let $z = e^{\frac{j4\pi}{7}}$. Draw $\{z, z^2, \dots, z^7\}$ in the complex plane.

16.1.1.9 Problem 9: Writing Vectors in Terms of Other Vectors

A pair of vectors $\mathbf{u} \in \mathbb{C}^2$ and $\mathbf{v} \in \mathbb{C}^2$ are called **linearly independent** if

$$\alpha\mathbf{u} + \beta\mathbf{v} = \mathbf{0} \quad \text{if and only if} \quad \alpha = \beta = 0$$

It is a fact that we can write any vector in \mathbb{C}^2 as a **weighted sum** (or **linear combination**) of any two linearly independent vectors, where the weights α and β are complex-valued.

⁵"Complex Numbers" <<http://cnx.org/content/m0081/latest/>>

⁶<http://www.dsp.rice.edu/courses/elec301/applets.shtml>

(a) Write $\begin{pmatrix} 3+4j \\ 6+2j \end{pmatrix}$ as a linear combination of $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $\begin{pmatrix} -5 \\ 3 \end{pmatrix}$. That is, find α and β such that

$$\begin{pmatrix} 3+4j \\ 6+2j \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \beta \begin{pmatrix} -5 \\ 3 \end{pmatrix}$$

(b) More generally, write $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ as a linear combination of $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $\begin{pmatrix} -5 \\ 3 \end{pmatrix}$. We will denote the answer for a given \mathbf{x} as $\alpha(x)$ and $\beta(x)$.

(c) Write the answer to (a) in matrix form, i.e. find a 2×2 matrix A such that

$$A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \alpha(x) \\ \beta(x) \end{pmatrix}$$

(d) Repeat (b) and (c) for a general set of linearly independent vectors \mathbf{u} and \mathbf{v} .

16.1.1.10 Problem 10: Fun with Fractals

A Julia set J is obtained by characterizing points in the complex plane. Specifically, let $f(x) = x^2 + \mu$ with μ complex, and define

$$g_0(x) = x$$

$$g_1(x) = f(g_0(x)) = f(x)$$

$$g_2(x) = f(g_1(x)) = f(f(x))$$

$$\vdots$$

$$g_n(x) = f(g_{n-1}(x))$$

Then for each x in the complex plane, we say $x \in J$ if the sequence

$$\{|g_0(x)|, |g_1(x)|, |g_2(x)|, \dots\}$$

does *not* tend to infinity. Notice that if $x \in J$, then each element of the sequence $\{g_0(x), g_1(x), g_2(x), \dots\}$ also belongs to J .

For most values of μ , the boundary of a Julia set is a fractal curve - it contains "jagged" detail no matter how far you zoom in on it. The well-known Mandelbrot set contains all values of μ for which the corresponding Julia set is connected.

(a) Let $\mu = -1$. Is $x = 1$ in J ?

(b) Let $\mu = 0$. What conditions on x ensure that x belongs to J ?

(c) Create an approximate picture of a Julia set in MATLAB. The easiest way is to create a matrix of complex numbers, decide for each number whether it belongs to J , and plot the results using the `imagesc` command. To determine whether a number belongs to J , it is helpful to define a limit N on the number of iterations of g . For a given x , if the magnitude $|g_n(x)|$ remains below some threshold M for all $0 \leq n \leq N$, we say that x belongs to J . The code below will help you get started:

```
N = 100;      % Max # of iterations
M = 2;       % Magnitude threshold
mu = -0.75;  % Julia parameter
realVals = [-1.6:0.01:1.6];
imagVals = [-1.2:0.01:1.2];

xVals = ones(length(imagVals),1) * realVals + ...
j*imagVals'*ones(1,length(realVals));

Jmap = ones(size(xVals));
g = xVals;   % Start with g0

% Insert code here to fill in elements of Jmap. Leave a '1'
% in locations where x belongs to J, insert '0' in the
% locations otherwise. It is not necessary to store all 100
% iterations of g!

imagesc(realVals, imagVals, Jmap);
colormap gray;
xlabel('Re(x)');
ylabel('Imag(x)');
```

This creates the following picture for $\mu = -0.75$, $N = 100$, and $M = 2$.

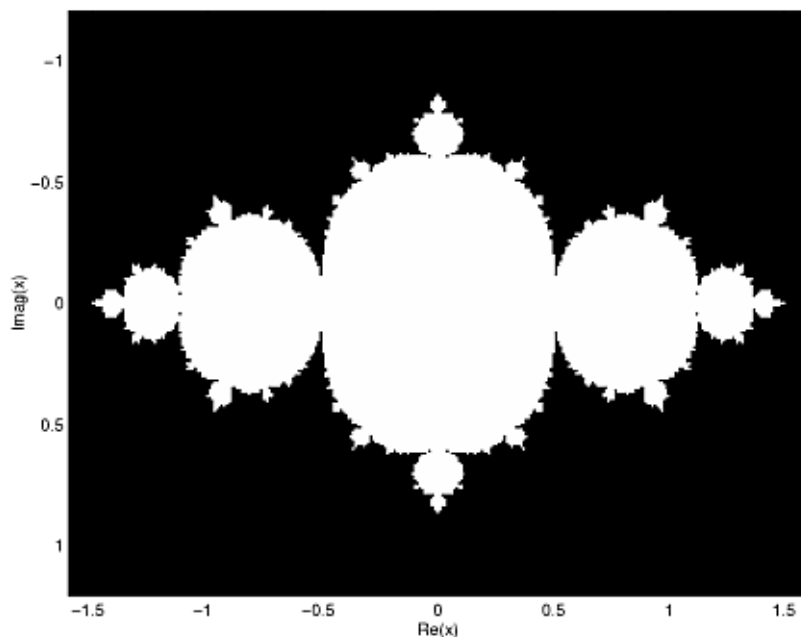


Figure 16.1: Example image where the x-axis is $Re(x)$ and the y-axis is $Im(x)$.

Using the *same* values for N , M , and x , create a picture of the Julia set for $\mu = -0.391 - 0.587j$. Print out this picture and hand it in with your MATLAB code.

JUST FOR FUN: Try assigning different color values to `Jmap`. For example, let `Jmap` indicate the first iteration when the magnitude exceeds M . Tip: try `imagesc(log(Jmap))` and `colormap jet` for a neat picture.

16.2 Homework #1 Solutions⁷

16.2.1 Problem #1

No solutions provided.

16.2.2 Problem #2

No solutions provided.

16.2.3 Problem #3

No solutions provided.

16.2.4 Problem #4

No solutions provided.

⁷This content is available online at <http://cnx.org/content/m10830/2.4/>.

16.2.5 Problem #5**16.2.5.1 Part (a)**

e^z lies on the unit circle for $z = \pm j$. When $z = \pm j$,

$$e^z = e^{\pm j} = \cos(\pm 1) + j \sin(\pm 1)$$

$$\begin{aligned} e^{\pm j} &= (\cos^2(\pm 1) + \sin^2(\pm 1))^{\frac{1}{2}} \\ &= 1 \end{aligned} \tag{16.1}$$

which gives us the unit circle!

Think of it this way: for $z = \sigma + j\theta$, you want a $\sigma = 0$ so that $e^{\sigma + j\theta}$ reduces as

$$e^{\sigma + j\theta} = e^\sigma e^{j\theta} = e^0 e^{j\theta} = e^{j\theta}$$

We know by Euler's formula (Section 1.6.2: Euler's Relation) that

$$e^{j\theta} = \cos(\theta) + j \sin(\theta)$$

The magnitude of this is given by $\sin^2(\theta) + \cos^2(\theta)$, which is 1 (which implies that $e^{j\theta}$ is on the unit circle).

So, we know we want to pick a $z = Aj\theta$ that is on the unit circle (from the problem statement), so we have to choose $A = \pm 1$ to get unit magnitude.

16.2.5.2 Part (b)

- $|\cdot|$ gives **magnitude** of complex number
- $Re(\cdot)$ gives **real part** of complex number
- $Im(\cdot)$ gives **imaginary part** of complex number

16.2.6 Problem #6**16.2.6.1 Part (a)**

$$\left(\frac{-1-j}{\sqrt{2}}\right)^{20} = \left(\frac{\sqrt{2}e^{\frac{5\pi}{4}}}{\sqrt{2}}\right)^{20} = \left(e^{\frac{5\pi}{4}}\right)^{20} = e^{j25\pi} = e^{j\pi} = -1$$

16.2.6.2 Part (b)

$$\frac{1+2j}{3+4j} = \left(\frac{1+2j}{3+4j}\right) \left(\frac{3-4j}{3-4j}\right) = \frac{3+6j-(4j+8)}{9+16} = \frac{11+2j}{25} = \frac{11}{25} + \frac{2}{25}j$$

16.2.6.3 Part (c)

$$\frac{1+\sqrt{3}j}{\sqrt{3}-j} = \frac{2e^{j\frac{\pi}{3}}}{2e^{j\frac{-\pi}{6}}} = e^{j\frac{\pi}{2}} = j$$

16.2.6.4 Part (d)

$$\sqrt{j} = (e^{j\frac{\pi}{2}})^{\frac{1}{2}} = e^{j\frac{\pi}{4}} = \cos\left(\frac{\pi}{4}\right) + j \sin\left(\frac{\pi}{4}\right) = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j$$

16.2.6.5 Part (e)

$$j^j = (e^{j\frac{\pi}{2}})^j = e^{j^2\frac{\pi}{2}} = e^{-\frac{\pi}{2}}$$

16.2.7 Problem #7**16.2.7.1 Part (a)**

$$z^2 - 4z = z(z - 4)$$

$$\text{Roots of } z = \{0, 4\}$$

16.2.7.2 Part (b)

$$z^2 - 4z + 4 = (z - 2)^2$$

$$\text{Roots of } z = \{2, 2\}$$

16.2.7.3 Part (c)

$$z^2 - 4z + 8$$

$$\text{Roots of } z = \frac{4 \pm \sqrt{16 - 32}}{2} = 2 \pm 2j$$

16.2.7.4 Part (d)

$$z^2 + 8$$

$$\text{Roots of } z = \frac{\pm\sqrt{-32}}{2} = \pm 2\sqrt{2}j$$

16.2.7.5 Part (e)

$$z^2 + 4z + 8$$

$$\text{Roots of } z = \frac{-4 \pm \sqrt{16 - 32}}{2} = -2 \pm 2j$$

16.2.7.6 Part (f)

$$2z^2 + 4z + 8$$

$$\text{Roots of } z = \frac{-4 \pm \sqrt{16 - 64}}{4} = -1 \pm \sqrt{3}j$$

16.2.7.7 Matlab Code and Plot

```
%%%%%%%%%%
%%%%% PROBLEM 7
%%%%%%%%%%
```

```
rootsA = roots([1 -4 0])
rootsB = roots([1 -4 4])
rootsC = roots([1 -4 8])
rootsD = roots([1 0 8])
```

```

rootsE = roots([1 4 8])
rootsF = roots([2 4 8])

zplane([rootsA; rootsB; rootsC; rootsD; rootsE; rootsF]);
gtext('a')
gtext('a')
gtext('b')
gtext('b')
gtext('c')
gtext('c')
gtext('d')
gtext('d')
gtext('e')
gtext('e')
gtext('f')
gtext('f')

```

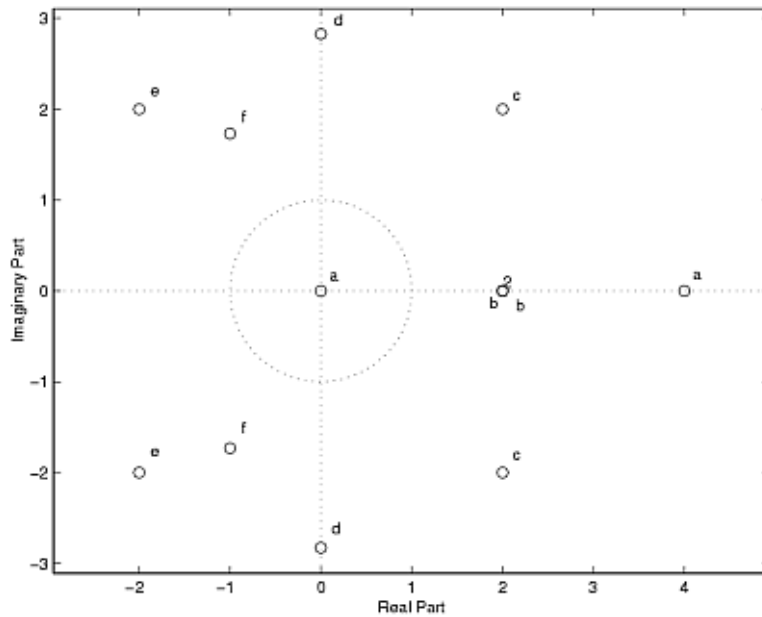


Figure 16.2: Plot of all the roots.

16.2.8 Problem #8

16.2.8.1 Part (a)

Raise $e^{\frac{j2\pi}{N}}$ to the N th power.

$$\left(e^{\frac{j2\pi}{N}}\right)^N = e^{j2\pi} = 1$$

NOTE: Similarly,

$$(1)^{\frac{1}{N}} = (e^{j2\pi})^{\frac{1}{N}} = e^{\frac{j2\pi}{N}}$$

16.2.8.2 Part (b)

For $z = e^{\frac{j2\pi}{7}}$,

$$z^k = \left(e^{\frac{j2\pi}{7}}\right)^k = e^{j2\pi\frac{k}{7}}$$

We will have points on the unit circle with angle of $\left\{\frac{2\pi}{7}, \frac{2\pi 2}{7}, \dots, \frac{2\pi 7}{7}\right\}$. The code used to plot these in MATLAB can be found below, followed by the plot.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% PROBLEM 8
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Part (b)

figure(1);
clf;
hold on;

th = [0:0.01:2*pi];
unitCirc = exp(j*th);
plot(unitCirc,'--');

for k = 1:7
    z = exp(j*2*pi*k/7);
    plot(z,'o');
    text(1.2*real(z),1.2*imag(z),strcat('z^',num2str(k)));
end

xlabel('real part');
ylabel('imag part');
title('Powers of exp(j2\pi/7) on the unit circle');
axis([-1.5 1.5 -1.5 1.5]);
axis square;

```

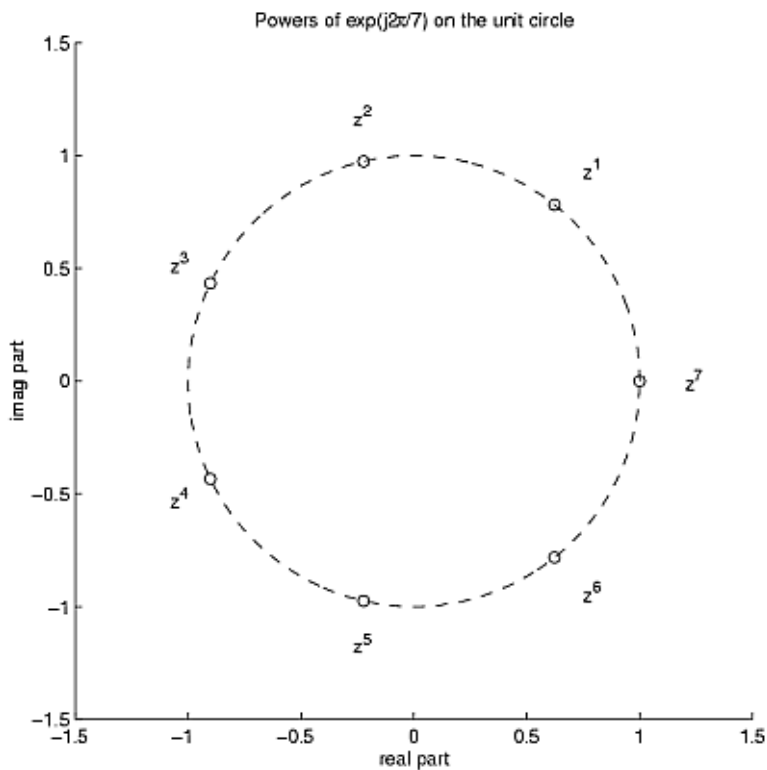


Figure 16.3: MATLAB plot of part (b).

16.2.8.3 Part (c)

For $z = e^{\frac{j4\pi}{7}}$,

$$z^k = \left(e^{\frac{j4\pi}{7}} \right)^k = e^{j2\pi \frac{2k}{7}}$$

Where we have

$$\{z, z^2, \dots, z^7\} = \left\{ e^{j2\pi \frac{2}{7}}, e^{j2\pi \frac{4}{7}}, e^{j2\pi \frac{6}{7}}, e^{j2\pi \frac{1}{7}}, e^{j2\pi \frac{3}{7}}, e^{j2\pi \frac{5}{7}}, 1 \right\}$$

The code used to plot these in MATLAB can be found below, followed by the plot.

```

%%% Part (c)

figure(1);
clf;
hold on;

th = [0:0.01:2*pi];
unitCirc = exp(j*th);

```



```

plot(unitCirc,'--');

for k = 1:7
    z = exp(j*4*pi*k/7);
    plot(z,'o');
    text(1.2*real(z),1.2*imag(z),strcat('z^',num2str(k)));
end

xlabel('real part');
ylabel('imag part');
title('Powers of exp(j4\pi/7) on the unit circle');
axis([-1.5 1.5 -1.5 1.5]);
axis square;

```

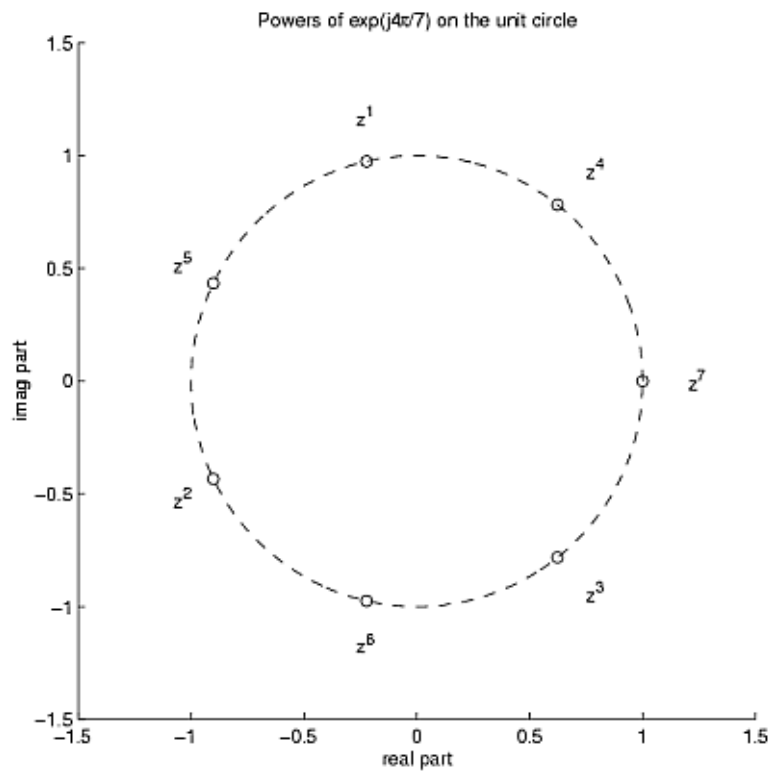


Figure 16.4: MATLAB plot of part (c).

16.2.9 Problem #9**16.2.9.1 Part (a)**

$$\begin{pmatrix} 3 + 4j \\ 6 + 2j \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \beta \begin{pmatrix} -5 \\ 3 \end{pmatrix}$$

To solve for β we must solve the following system of equations:

$$\alpha - 5\beta = 3 + 4j$$

$$2\alpha + 3\beta = 6 + 2j$$

If we multiply the top equation by -2 we will get the following, which allows us to cancel out the alpha terms:

$$-2\alpha + 10\beta = -6 - 8j$$

$$2\alpha + 3\beta = 6 + 2j$$

And now we have,

$$13\beta = -6j$$

$$\beta = \frac{-6}{13}j$$

And to solve for α we have the following equation:

$$\begin{aligned} \alpha &= 3 + 4j + 5\beta \\ &= 3 + 4j + 5\left(\frac{-6}{13}j\right) \\ &= 3 + \frac{22}{13}j \end{aligned} \tag{16.2}$$

16.2.9.2 Part (b)

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \beta \begin{pmatrix} -5 \\ 3 \end{pmatrix}$$

$$x_1 = \alpha - 5\beta$$

$$x_2 = 2\alpha + 3\beta$$

Solving for α and β we get:

$$\alpha(x) = \frac{3x_1 + 5x_2}{13}$$

$$\beta(x) = \frac{-2x_1 + x_2}{13}$$

16.2.9.3 Part (c)

$$\begin{pmatrix} \alpha(x) \\ \beta(x) \end{pmatrix} = \begin{pmatrix} \frac{3}{13} & \frac{5}{13} \\ \frac{-2}{13} & \frac{1}{13} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

16.2.9.4 Part (d)

Write $u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$ and $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$. Then solve

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \alpha \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \beta \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

which corresponds to the system of equations

$$x_1 = \alpha u_1 + \beta v_1$$

$$x_2 = \alpha u_2 + \beta v_2$$

Solving for α and β we get

$$\alpha(x) = \frac{v_2 x_1 - v_1 x_2}{u_1 v_2 - u_2 v_1}$$

$$\beta(x) = \frac{u_2 x_1 - u_1 x_2}{v_1 u_2 - v_2 u_1}$$

For the matrix A we get

$$A = \frac{1}{u_1 v_2 - u_2 v_1} \begin{pmatrix} v_2 & -v_1 \\ -u_2 & u_1 \end{pmatrix}$$

16.2.10 Problem #10**16.2.10.1 Part (a)**

If $u = -1$, then $f(x) = x^2 - 1$. Examine the sequence $\{g_0(x), g_1(x), \dots\}$:

$$g_0(x) = 1$$

$$g_1(x) = 1^2 - 1 = 0$$

$$g_2(x) = 0^2 - 1 = -1$$

$$g_3(x) = (-1)^2 - 1 = 0$$

$$g_4(x) = 0^2 - 1 = -1$$

⋮

The magnitude sequence remains bounded so $x = 1$ belongs to J .

16.2.10.2 Part (b)

If $u = 0$, then $f(x) = x^2$. So we have

$$g_0(x) = x$$

$$g_1(x) = x^2$$

$$g_2(x) = (x^2)^2 = x^4$$

$$\vdots$$

$$g_n(x) = (x^2)^n = x^{2n}$$

Writing $x = re^{j\theta}$, we have $g_n(x) = x^{2n} = r^{2n}e^{j\theta 2n}$, and so we have

$$|g_n(x)| = r^{2n}$$

The magnitude sequence blows up if and only if $r > 1$. Thus x belongs to J if and only if $|x| \leq 1$. So, J corresponds to the unit disk.

16.2.10.3 Part (c)

```
%%%%%%%%%%
%%% PROBLEM 10
%%%%%%%%%%
```

```
%%% Part (c) - solution code
```

```
N = 100;           % Max # of iterations
M = 2;            % Magnitude threshold
```

```
mu = -0.391 - 0.587*j; % Julia parameter
realVals = [-1.6:0.01:1.6];
imagVals = [-1.2:0.01:1.2];
xVals = ones(length(imagVals),1)*realVals + ...
j*imagVals'*ones(1,length(realVals));
Jmap = ones(size(xVals));
g = xVals; % Start with g0
```

```
for n = 1:N
g = g.^2 + mu;
big = (abs(g) > M);
Jmap = Jmap.*(1-big);
end
```

```
imagesc(realVals,imagVals,Jmap); colormap gray;
xlabel('Re(x)'); ylabel('Imag(x)');
```

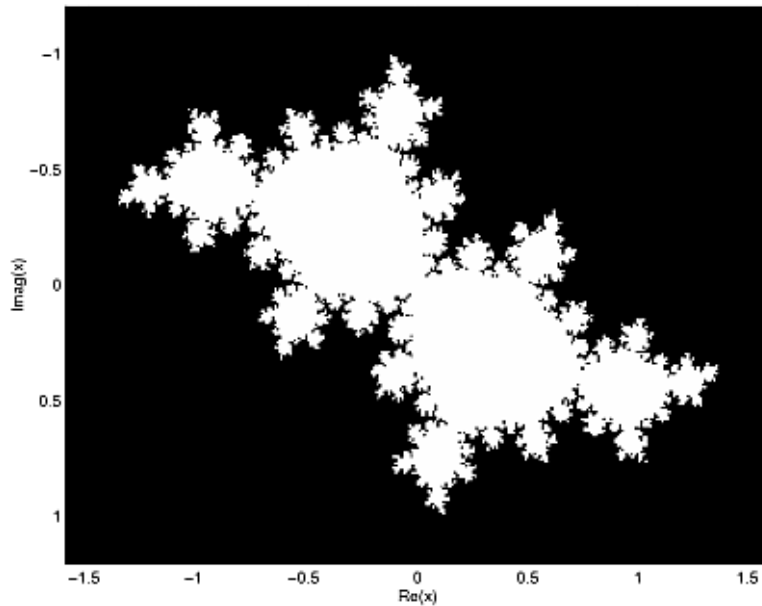


Figure 16.5: MATLAB plot of part (c).

16.2.10.4 Just for Fun Solution

```

%%% Just for fun code

N = 100;           % Max # of iterations
M = 2;           % Magnitude threshold
mu = -0.391 - 0.587*j; % Julia parameter
realVals = [-1.6:0.005:1.6];
imagVals = [-1.2:0.005:1.2];

xVals = ones(length(imagVals),1)*realVals + ...
j*imagVals'*ones(1,length(realVals));
Jmap = zeros(size(xVals));

% Now, we put zeros in the 'middle', for a
% cool effect.
g = xVals; % Start with g0

for n = 1:N
    g = g.^2 + mu;
    big = (abs(g) > M);
    notAlreadyBig = (Jmap == 0);
    Jmap = Jmap + n*(big.*notAlreadyBig);

```

```
end  
  
imagesc(realVals,imagVals,log(Jmap));colormap jet;  
xlabel('Re(x)'); ylabel('Imag(x)');
```

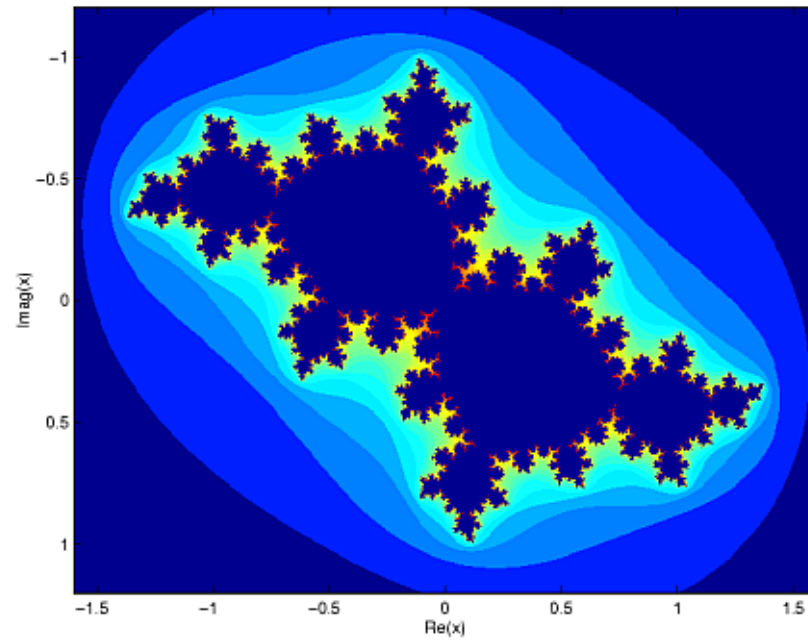


Figure 16.6: MATLAB plot.

Chapter 17

Viewing Embedded LabVIEW Content¹

In order to view LabVIEW content embedded in Connexions modules, you must install the LabVIEW Run-time Engine on your computer. The following are sets of instructions for installing the software on different platforms.

NOTE: Embedded LabVIEW content is currently supported only under Windows 2000/XP. Also, you must have version 8.0.1 of the LabView Run-time Engine to run much of the embedded content in Connexions.

17.1 Installing the LabVIEW Run-time Engine on Microsoft Windows 2000/XP

1. Point your web browser to the LabVIEW Run-time Engine download page at: <http://digital.ni.com/softlib.nsf/websearch/077b51e8d15604bd8625711c006240e7>².
2. If you're not logged in to NI, click the link to continue the download process at the bottom of the page.
3. Login or create a profile with NI to continue.
4. Once logged in, click the `LabVIEW_8.0.1_Runtime_Engine.exe` link and save the file to disk.
5. Once the file has downloaded, double click it and follow the steps to install the run-time engine.
6. Download the LabVIEW Browser Plug-in at: <http://zone.ni.com/devzone/concept.d.nsf/webmain/7DBFD404C6AD0B2>³.
7. Put the `LVBrowserPlugin.ini` file in the `My Documents\LabVIEW Data` folder. (You may have to create this folder if it doesn't already exist.)
8. Restart your web browser to complete the installation of the plug-in.

¹This content is available online at <http://cnx.org/content/m13753/1.3/>.

²<http://digital.ni.com/softlib.nsf/websearch/077b51e8d15604bd8625711c006240e7>

³[http://zone.ni.com/devzone/concept.d.nsf/webmain/7DBFD404C6AD0B24862570BB0072F83B/\\$FILE/LVBrowserPlugin.ini](http://zone.ni.com/devzone/concept.d.nsf/webmain/7DBFD404C6AD0B24862570BB0072F83B/$FILE/LVBrowserPlugin.ini)

Glossary

B Basis

A basis for \mathbb{C}^n is a set of vectors that: (1) spans \mathbb{C}^n and (2) is linearly independent.

C Cauchy-Schwarz Inequality

For \mathbf{x}, \mathbf{y} in an inner product space

$$| \langle \mathbf{x}, \mathbf{y} \rangle | \leq \| \mathbf{x} \| \| \mathbf{y} \|$$

with equality holding *if and only if* \mathbf{x} and \mathbf{y} are linearly dependent (Section 5.1.1: Linear Independence), i.e. $\mathbf{x} = \alpha \mathbf{y}$ for some scalar α .

D difference equation

An equation that shows the relationship between consecutive values of a sequence and the differences among them. They are often rearranged as a recursive formula so that a systems output can be computed from the input signal and past outputs.

Example:

$$y[n] + 7y[n-1] + 2y[n-2] = x[n] - 4x[n-1] \quad ()$$

domain

The group, or set, of values that are defined by a given function.

Example: Using the rational function above, (14.38), the domain can be defined as any real number x where x does not equal 1 or negative 3. Written out mathematically, we get the following:

$$\{ x \in \mathbb{R} \mid x \neq -3 \text{ and } x \neq 1 \} \quad ()$$

E eigenvector

An eigenvector of A is a vector $\mathbf{v} \in \mathbb{C}^n$ such that

$$A\mathbf{v} = \lambda \mathbf{v} \quad (5.2)$$

where λ is called the corresponding **eigenvalue**. A only changes the *length* of \mathbf{v} , not its direction.

I inner product

The inner product is defined mathematically as:

$$\begin{aligned} \langle \mathbf{x}, \mathbf{y} \rangle &= \mathbf{y}^T \mathbf{x} \\ &= \begin{pmatrix} y_0 & y_1 & \cdots & y_{n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} \\ &= \sum_{i=0}^{n-1} (x_i y_i) \end{aligned} \quad (15.1)$$

L

$$l_p [0, N - 1] = \{f \in \mathbb{C}^N, \|f\|_p < \infty\}$$

but from previous discussion $l_p [0, N - 1] = \mathbb{C}^N$

$$L_p (R) = \{f, \|f\|_p < \infty\}$$

$$\|f\|_p = \left(\int_{-\infty}^{\infty} (|f[n]|)^p dt \right)^{\frac{1}{p}}$$

where $1 \leq p < \infty$

$$\|f\|_{\infty} = \text{esssup}|f(t)|$$

where $-\infty < t < \infty$

$$L_p [T_1, T_2] = \{f [T_1, T_2], \|f\|_p < \infty\}$$

$$\|f\|_p = \left(\int_{T_1}^{T_2} (|f(t)|)^p dt \right)^{\frac{1}{p}}$$

where $1 \leq p < \infty$

$$\|f\|_p = \text{esssup}|f(t)|$$

where $T_1 \leq t \leq T_2$

$$l_p (z) = \{f, \|f\|_p < \infty\}$$

$$\left(\|f\|_p \right)^p = \sum_{n=-\infty}^{\infty} (|f[n]|)^p$$

where $1 \leq p < \infty$

$$\|f\|_{\infty} = \max_{n \in z} \{|f[n]|\}$$

limit

A sequence $\{g_n\}_{n=1}^{\infty}$ converges to a limit $g \in \mathbb{R}$ if for every $\epsilon > 0$ there is an integer N such that

$$|g_i - g| < \epsilon, \quad i \geq N$$

We usually denote a limit by writing

$$\lim_{i \rightarrow \infty} g_i = g$$

or

$$g_i \rightarrow g$$

Linearly Independent

For a given set of vectors, $\{x_1, x_2, \dots, x_n\}$, they are linearly independent if

$$c_1x_1 + c_2x_2 + \dots + c_nx_n = 0$$

only when $c_1 = c_2 = \dots = c_n = 0$

Example: We are given the following two vectors:

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$x_2 = \begin{pmatrix} -6 \\ -4 \end{pmatrix}$$

These are *not linearly independent* as proven by the following statement, which, by inspection, can be seen to not adhere to the definition of linear independence stated above.

$$x_2 = -2x_1 \Rightarrow 2x_1 + x_2 = 0$$

Another approach to reveal a vectors independence is by graphing the vectors. Looking at these two vectors geometrically (as in Figure 5.1), one can again prove that these vectors are *not* linearly independent.

N Normalized Basis

a basis (Section 5.1.3: Basis) $\{b_i\}$ where each b_i has unit norm

$$\|b_i\| = 1, \quad i \in \mathbb{Z} \tag{15.19}$$

O Orthogonal Basis

a basis $\{b_i\}$ in which the elements are **mutually orthogonal**

$$\langle b_i, b_j \rangle = 0, \quad i \neq j$$

orthogonal

We say that \mathbf{x} and \mathbf{y} are orthogonal if:

$$\langle \mathbf{x}, \mathbf{y} \rangle = 0$$

Orthonormal Basis

a basis that is both **normalized** and **orthogonal**

$$\|b_i\| = 1, \quad i \in \mathbb{Z}$$

$$\langle b_i, b_j \rangle = 0, \quad i \neq j$$

P poles

Also called singularities, these are the points s at which $\mathcal{L}_{x_1}(s)$ blows up.

poles

1. The value(s) for z where $Q(z) = 0$.

2. The complex frequencies that make the overall gain of the filter transfer function infinite.

poles

1. The value(s) for z where the *denominator* of the transfer function equals zero
2. The complex frequencies that make the overall gain of the filter transfer function infinite.

R rational function

For any two polynomials, A and B, their quotient is called a rational function.

Example: Below is a simple example of a basic rational function, $f(x)$. Note that the numerator and denominator can be polynomials of any order, but the rational function is undefined when the denominator equals zero.

$$f(x) = \frac{x^2 - 4}{2x^2 + x - 3} \quad ()$$

S sequence

A sequence is a function g_n defined on the positive integers 'n'. We often denote a sequence by $\{g_n\}_{n=1}^{\infty}$

Example: A real number sequence:

$$g_n = \frac{1}{n}$$

Example: A vector sequence:

$$g_n = \begin{pmatrix} \sin\left(\frac{n\pi}{2}\right) \\ \cos\left(\frac{n\pi}{2}\right) \end{pmatrix}$$

Example: A function sequence:

$$g_n(t) = \begin{cases} 1 & \text{if } 0 \leq t < \frac{1}{n} \\ 0 & \text{otherwise} \end{cases}$$

NOTE: A function can be thought of as an infinite dimensional vector where for each value of 't' we have one dimension

Span

The span⁴ of a set of vectors $\{x_1, x_2, \dots, x_k\}$ is the set of vectors that can be written as a linear combination of $\{x_1, x_2, \dots, x_k\}$

$$\text{span}(\{x_1, \dots, x_k\}) = \{\alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_k x_k \ , \ \alpha_i \in \mathbb{C}^n \}$$

Example: Given the vector

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

the span of x_1 is a *line*.

⁴"Subspaces", Definition 2: "Span" <<http://cnx.org/content/m10297/latest/#defn2>>

Example: Given the vectors

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$x_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

the span of these vectors is \mathbb{C}^2 .

U Uniform Convergence

The sequence (Section 9.1) $\{g_n\}_{n=1}^{\infty}$ converges uniformly to function g if for every $\epsilon > 0$ there is an integer N such that $n \geq N$ implies

$$|g_n(t) - g(t)| \leq \epsilon \tag{9.7}$$

for all $t \in \mathbb{R}$.

V Vector space

A linear vector space S is a collection of "vectors" such that (1) if $f_1 \in S \Rightarrow \alpha f_1 \in S$ for all scalars α (where $\alpha \in \mathbb{R}$ or $\alpha \in \mathbb{C}$) and (2) if $f_1 \in S$, $f_2 \in S$, then $f_1 + f_2 \in S$

Z zeros

1. The value(s) for z where $P(z) = 0$.
2. The complex frequencies that make the overall gain of the filter transfer function zero.

zeros

1. The value(s) for z where the *numerator* of the transfer function equals zero
2. The complex frequencies that make the overall gain of the filter transfer function zero.

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

- A** alias, § 12.5(209), § 12.6(212)
 aliasing, § 7.2(142), § 12.5(209), 209,
 § 12.6(212)
 almost everywhere, § 6.11(134)
 alphabet, § 1.7(31), 34
 analog, § 1.1(1), 2, 31, § 10.6(185), § 10.7(186)
 analysis, 114
 Anti-Aliasing, § 12.6(212)
 anticausal, § 1.1(1), 3
 aperiodic, § 1.1(1), 112
 approximation, § 15.13(298)
- B** bandlimited, 199
 baraniuk, § 16.2(307)
 bases, § 5.1(91)
 basis, § 5.1(91), 94, 94, 94, § 7.2(142), 147,
 § 7.3(149), 205, § 15.7(280), § 15.8(283), 284,
 § 15.9(287), § 15.10(288), § 15.11(295)
 basis matrix, § 15.8(283), 285
 best approximates, 299
 BIBO, § 3.4(65)
 bilateral Laplace transform pair, 219
 bilateral z-transform, 231
 bounded input bounded output, § 3.4(65)
 bounded input-bounded output (BIBO), 40
 boxcar filter, 72
 butterfly, § 8.3(167), 169
- C** Cartesian, 304
 cascade, § 2.2(41)
 cauch-schwarz, § 15.5(270)
 cauchy, § 15.5(270)
 cauchy-schwarz inequality, § 15.5(270), 271
 causal, § 1.1(1), 3, § 2.1(37), 39, § 2.2(41), 257
 characteristic equation, 52
 characteristic polynomial, 253
 circular, § 4.3(158), § 6.7(126), § 4.3(158)
 circular convolution, § 4.3(158), 78, § 6.7(126),
 § 4.3(158), 159
 circular shift, § 7.5(154)
 circular shifting, 156
 circular shifts, § 7.5(154), 154
 coefficient, § 6.2(112)
 coefficient vector, § 15.8(283), 285
 commutative, 53, 60, 73
 complex, § 1.7(31), § 7.2(142), § 14.7(254)
 complex amplitude, 29
 complex continuous-time exponential signal,
 29
 complex exponential, 23, § 1.6(28), 28
 complex exponential sequence, 32
 complex plane, § 1.6(28)
 complex sinusoids, § 7.2(142)
 complex vector space, 263
 complex vector spaces, § 15.1(263)
 complex-valued, § 1.7(31)
 complex-valued function, 220, 220
 complexity, 165, § 8.3(167)
 composite, 169
 computational advantage, 167
 conjugates, 147
 Constant-Coefficient, § 4.4(82)
 continuous, 1, 132
 continuous frequency, § 10.4(184), § 11.1(191)
 continuous system, 37
 continuous time, § 1.1(1), § 1.4(22), § 3.1(47),
 § 3.2(53), § 3.4(65), § 7.1(141), § 10.3(183),
 § 11.1(191), § 11.2(192), § 12.1(197), 197,
 § 12.7(214), § 13.1(219), § 13.2(222),
 § 13.3(223), § 13.4(223), § 13.6(227)
 continuous-time, § 3.1(47)
 Continuous-Time Fourier Transform, 191
 control theory, 229
 converge, § 6.9(130), § 9.3(176)
 convergence, § 6.9(130), 132, § 9.1(171),
 § 9.2(173), § 9.3(176)
 converges, § 9.3(176)
 convolution, § 3.2(53), § 3.3(59), § 4.2(73), 73,
 § 4.3(158), § 6.7(126), § 4.3(158), § 11.2(192)
 convolution integral, 53
 convolution sum, 73
 convolutions, § 4.3(158), § 4.3(158)
 convolve, § 4.3(158), § 4.3(158)

- Cooley-Tukey, § 8.1(165), § 8.3(167)
 csi, § 15.5(270)
 CTFT, § 11.1(191), § 12.1(197)
 cuachy, § 15.5(270)
- D** de, § 3.1(47)
 Decaying Exponential, 23
 decompose, § 1.7(31), § 15.8(283), 284
 delta function, § 15.7(280)
 determinant, § 5.2(96)
 deterministic signal, 7
 dft, § 4.3(158), § 7.5(154), § 4.3(158),
 § 8.2(166), § 10.1(179), § 10.2(181)
 difference equation, § 4.1(69), 69, 82, 251, 251
 Difference Equations, § 4.4(82), § 4.5(83)
 differential, § 3.1(47)
 differential equations, § 3.1(47)
 digital, § 1.1(1), 2, § 10.6(185), § 10.7(186)
 digital signal processing, § 4.1(69), § 10.7(186)
 Dirac delta, 23
 dirac delta function, § 1.4(22), § 1.5(25), 25
 direct method, 253
 dirichlet conditions, § 6.10(132), 132
 Dirichlet sinc, § 10.1(179)
 discontinuity, § 6.9(130), 132
 discontinuous functions, 135
 discrete, 1, § 7.2(142), § 12.7(214)
 discrete fourier transform, § 4.3(158),
 § 4.3(158), § 10.2(181), 181
 discrete system, 37
 discrete time, § 1.1(1), § 3.4(65), § 4.2(73),
 § 7.1(141), § 7.5(154), § 10.4(184), § 12.1(197),
 § 13.6(227), § 14.2(236)
 discrete time fourier series, § 7.2(142), 142,
 § 7.3(149), § 7.4(150)
 discrete time processing, § 12.7(214)
 discrete time signals, 197
 discrete-time, § 1.7(31), § 7.3(149),
 § 10.5(185), § 10.6(185), § 10.7(186)
 discrete-time convolution, 73
 discrete-time exponential signal, 29
 discrete-time Fourier transform, § 10.7(186)
 Discrete-Time Fourier Transform properties,
 § 10.5(185)
 discrete-time sinc function, 189
 discrete-time systems, § 4.1(69)
 domain, 250, 250
 dot product, § 15.3(268)
 dot products, 268
 DSP, § 4.1(69), § 10.7(186), § 12.4(207),
 § 14.2(236)
 DT, § 4.2(73)
 dtfs, § 7.2(142), § 7.3(149), § 7.4(150)
 DTFT, § 10.1(179), § 10.4(184), § 12.1(197)
 dynamic content, § 17(319)
- E** edgy, 118
 eigen, § 5.4(104)
 eigenfunction, § 5.2(96), § 5.4(104), § 5.5(105),
 106, § 6.4(116)
 eigenfunctions, § 5.4(104), § 6.2(112),
 § 6.4(116)
 eigensignal, 106
 eigenvalue, § 5.2(96), 320, 97, § 5.4(104),
 § 5.5(105)
 eigenvalues, § 5.2(96), § 5.4(104)
 eigenvector, § 5.2(96), 96, § 5.4(104),
 § 5.5(105)
 eigenvectors, § 5.4(104)
 Elec 301, § 1.2(9), § 4.4(82), § 4.5(83),
 § 16.1(303), § 16.2(307)
 elec301, § 16.1(303), § 16.2(307)
 embedded, § 17(319)
 energy, 131
 Euclidean norm, § 15.2(265)
 Euler's Identity, 29
 Euler's Relation, 30
 even signal, § 1.1(1), 4, § 6.6(122)
 example, § 10.7(186)
 examples, § 10.7(186)
 existence, 132
 expansion, § 15.9(287), § 15.10(288)
 exponential, § 1.4(22), § 1.7(31), § 6.3(115)
 exponential function, 28
- F** fast Fourier transform, § 8.1(165), § 8.2(166),
 § 8.3(167)
 FFT, § 8.1(165), § 8.2(166), § 8.3(167),
 § 10.1(179)
 filter, § 12.6(212), § 14.8(258)
 finite-duration sequence, 237
 finite-length sequence, 248
 finite-length signal, 8
 FIR, 72
 form, 167
 fourier, § 5.2(96), § 6.2(112), § 6.3(115),
 § 6.4(116), § 6.5(119), § 6.6(122), § 6.7(126),
 § 6.8(127), § 6.9(130), § 6.10(132), § 6.12(137),
 § 7.2(142), § 7.3(149), § 7.4(150), § 7.5(154),
 § 8.2(166), § 15.10(288)
 fourier analysis, § 7.2(142)
 fourier coefficient, § 6.9(130)
 fourier coefficients, § 6.2(112), 113, § 6.3(115)
 fourier series, § 5.2(96), § 6.2(112), § 6.3(115),

- § 6.4(116), § 6.6(122), § 6.7(126), § 6.9(130), § 6.10(132), § 6.11(134), § 6.12(137), § 7.1(141), § 7.2(142), § 7.3(149), § 7.4(150), § 15.7(280), § 15.10(288)
 - fourier transform, § 4.3(158), § 6.10(132), § 7.1(141), § 7.5(154), § 4.3(158), § 8.2(166), § 10.2(181), § 10.3(183), § 10.4(184), § 10.5(185), § 10.6(185), § 10.7(186), § 11.1(191), § 11.2(192), § 14.1(231), 231
 - fourier transform pairs, § 5.6(108)
 - frequency, § 10.6(185)
 - Frequency Domain, § 10.5(185)
 - frequency shift keying, § 15.5(270)
 - fsk, § 15.5(270)
 - function, § 14.5(249)
 - function sequences, § 9.3(176)
 - function space, § 15.9(287)
 - function spaces, 95, § 15.9(287)
 - fundamental period, 2
- G**
- geometric series, 186
 - Gibb's phenomenon, 137
 - gibbs phenomenon, § 6.11(134), 135
 - graphical method, 56
 - Growing Exponential, 23
- H**
- haar, § 15.10(288)
 - haar wavelet, § 15.10(288)
 - harmonic, § 7.2(142)
 - harmonic sinusoids, § 7.2(142), 143
 - hermitian, § 15.11(295)
 - hilbert, § 15.4(270), § 15.5(270), § 15.6(277), § 15.8(283), § 15.10(288)
 - Hilbert space, 270, § 15.6(277), § 15.7(280), § 15.13(298)
 - hilbert spaces, § 15.4(270), § 15.6(277), § 15.8(283), § 15.10(288)
 - homework 1, § 16.1(303)
 - homework one, § 16.1(303)
 - homogeneous solution, 253
- I**
- identity matrix, 286
 - IIR, 71
 - imaginary part, 308
 - Important note:, 216
 - impulse, § 1.4(22), § 1.5(25)
 - impulse response, 28, § 4.2(73)
 - independence, § 5.1(91)
 - indirect method, 253
 - infinite-length signal, 8
 - initial conditions, 70, 251
 - inner, § 15.4(270)
 - inner product, § 15.3(268), 268, § 15.4(270), § 15.5(270), § 15.11(295)
 - inner product space, 270
 - inner products, § 15.3(268), 268, § 15.5(270)
 - interpolation, § 12.2(201)
 - Inverse Laplace Transform, § 13.5(225)
 - inverse transform, § 6.2(112), 114
- K**
- kroncker, § 15.7(280)
- L**
- LabVIEW, § 17(319)
 - laplace transform, § 3.4(65), § 7.1(141), § 13.1(219), § 13.2(222), § 13.3(223), § 13.4(223), § 13.6(227)
 - left-handed, 7
 - limit, 171
 - linear, § 2.1(37), 37, § 2.2(41), § 4.4(82)
 - linear algebra, § 5.1(91)
 - linear and time-invariant, 73
 - linear combination, 304
 - linear convolution, 78, 159
 - linear function space, 263
 - linear function spaces, § 15.1(263)
 - linear independence, § 5.1(91)
 - linear system, § 3.1(47), § 5.2(96)
 - linear time invariant, § 3.2(53), § 5.5(105)
 - linear time-invariant systems, 247
 - linear transformation, § 6.5(119), 120
 - linearity, § 11.2(192)
 - linearly independent, 91, 91, 304
 - LTI, § 3.2(53), 73, § 5.4(104), § 5.5(105), § 6.2(112), § 6.8(127)
 - LTI system, § 6.2(112), § 6.4(116)
- M**
- magnitude, 308
 - matched filter, § 15.5(270)
 - matched filter detector, § 15.5(270), 271
 - matched filters, § 15.5(270)
 - matrix equation, § 7.3(149)
 - maxima, 133
 - maximum, 269
 - mean square, 137
 - minima, 133
 - modulation, § 11.2(192)
 - mutually orthogonal, 322
- N**
- noisy signals, 118
 - nonanticipative, 39
 - noncausal, § 1.1(1), 3, § 2.1(37), 39
 - nonlinear, § 2.1(37), 37
 - nonuniform convergence, § 6.11(134), 135
 - Norm, § 1.2(9), § 9.1(171), § 9.2(173), § 15.2(265), 265, § 15.3(268)

- norm convergence, § 9.1(171), § 9.2(173)
 normalization, § 15.2(265)
 normalized, § 15.7(280), 322
 normalized basis, § 15.7(280), 280, 280
 normed linear space, § 15.2(265), 265, 270
 normed space, § 15.6(277)
 normed vector space, § 15.2(265), 265
 norms, § 15.2(265)
 not, 166
 Nth Root of Unity, 304
 Nyquist, § 10.6(185), § 12.4(207)
 Nyquist frequency, § 10.7(186), § 12.4(207), 207
 Nyquist theorem, § 12.4(207), 207
- O** odd signal, § 1.1(1), 4, § 6.6(122)
 on our computer!!!, 216
 order, § 8.3(167), § 13.5(225), 227, 251
 orthogonal, § 15.3(268), 269, 269, § 15.7(280), 322, 299
 orthogonal basis, § 15.7(280), 280
 orthonormal, § 7.2(142), § 15.7(280), § 15.8(283)
 orthonormal basis, § 7.2(142), 148, § 15.7(280), 281, § 15.8(283), 283
- P** parallel, § 2.2(41)
 Parseval, § 15.12(297)
 Parseval's Theorem, § 10.7(186)
 particular solution, 253
 perfect, § 12.3(205)
 period, 2, § 6.1(111), 111, § 7.4(150)
 periodic, § 1.1(1), § 6.1(111), § 7.4(150)
 periodic function, § 6.1(111), 111
 periodicity, § 6.1(111)
 phasor, 29
 Plancharel, § 15.12(297)
 point wise, § 9.1(171), § 9.2(173)
 pointwise, § 6.9(130), 137, § 9.1(171), § 9.2(173), 173
 pointwise convergence, § 6.9(130), § 9.2(173)
 pole, § 3.4(65), § 13.4(223), § 13.6(227), § 14.7(254), § 14.8(258)
 pole-zero cancellation, 229
 poles, § 13.5(225), 226, 227, 250, 254
 polynomial, § 14.5(249)
 power series, 232, 237
 processing, § 12.7(214)
 projection, § 15.3(268), 269, § 15.13(298)
 properties, § 6.6(122)
 property, § 3.3(59)
 proportional, 165
- R** random signal, 7
 rational, § 14.5(249)
 rational function, § 14.5(249), 249, 249
 rational functions, § 14.5(249)
 real part, 308
 real vector space, 263, 276
 real vector spaces, § 15.1(263)
 real-valued, § 1.7(31)
 reconstruct, § 12.3(205)
 reconstruction, § 12.2(201), § 12.3(205), § 12.4(207), § 12.5(209)
 region of convergence, § 13.4(223)
 region of convergence (ROC), 223
 rest, 87
 right-handed, 7
 right-sided sequence, 239, 240
 ROC, § 13.4(223), § 14.1(231), 232, 237
 root mean squared, 17
- S** s-plane, 31
 sample, § 12.4(207), § 12.5(209)
 sampling, 181, 197, § 12.3(205), § 12.4(207), § 12.5(209), § 12.6(212)
 schwarz, § 15.5(270)
 sequence, 171
 sequence of functions, 131
 Sequence-Domain, § 10.5(185)
 sequences, § 1.7(31), § 9.1(171), § 9.3(176)
 shift-invariant, 69
 shift-invariant systems, § 4.1(69)
 sifting property, § 1.4(22), § 1.5(25), 27
 signal, § 6.1(111), § 6.12(137)
 signals, § 1.3(19), § 1.4(22), § 1.5(25), § 1.6(28), § 1.7(31), § 2.1(37), § 3.2(53), § 3.3(59), § 3.4(65), § 4.2(73), § 6.2(112), § 6.3(115), § 7.1(141), § 10.3(183), § 13.4(223), § 13.6(227)
 signals and systems, § 1.1(1), § 4.2(73)
 sinc, § 12.3(205)
 sine, § 1.7(31)
 singularities, § 13.5(225), 225
 sinusoid, § 1.7(31), § 6.3(115)
 smooth signals, 118
 span, § 5.1(91), 93, § 5.4(104), 287
 square pulse, 135
 stability, § 3.4(65)
 stable, 40, 257
 standard basis, 94, § 7.3(149), § 15.8(283)
 strong dirichlet condition, § 6.10(132)
 Strong Dirichlet Conditions, 133
 superposition, § 2.2(41), § 4.1(69)
 symbolic-valued signals, § 1.7(31)

- symmetry, § 6.5(119), § 6.6(122), § 11.2(192)
 - symmetry properties, § 6.6(122)
 - symmetry property, § 6.6(122)
 - synthesis, 114
 - system, § 4.3(158), § 5.4(104), § 6.4(116), § 6.12(137), § 4.3(158)
 - systems, § 1.7(31), § 3.2(53), § 3.3(59), § 3.4(65), § 7.1(141), § 10.3(183), § 13.4(223), § 13.6(227)
- T**
- t-periodic, § 6.1(111)
 - threshold, 273
 - time differentiation, § 11.2(192)
 - time domain, § 4.1(69)
 - time invariant, § 2.1(37), 38
 - time reversal, § 1.3(19)
 - time scaling, § 1.3(19), § 11.2(192)
 - time shifting, § 1.3(19), § 11.2(192)
 - time variant, 39
 - time varying, § 2.1(37)
 - time-invariant, § 2.2(41)
 - transfer function, 252
 - transform, § 6.2(112), 114
 - transform pairs, § 14.2(236)
 - transforms, 102, 287
 - transpose, § 15.11(295)
 - two-sided sequence, 241
- U**
- uniform, § 9.3(176)
 - uniform convergence, § 9.3(176), 176
 - unilateral, § 14.2(236)
 - unilateral z-transform, 231
 - unique, 284
 - unit impulse, 23
 - unit sample, § 1.7(31), 33
- unit step, § 1.4(22)
 - unit-step function, 24
 - unity, 25
 - unstable, 41
- V**
- vector, § 9.2(173), § 15.1(263)
 - vector space, § 15.1(263), 263, § 15.6(277), § 15.7(280), § 15.9(287)
 - vector spaces, § 15.1(263), § 15.6(277), § 15.7(280)
 - vectors, § 9.2(173)
 - vertical asymptotes, 250
 - VI, § 17(319)
 - virtual instrument, § 17(319)
- W**
- wavelet, § 15.10(288)
 - wavelets, § 15.10(288), 288
 - weak dirichlet condition, § 6.10(132), 133
 - weighted sum, 304
- X**
- x-intercept, 250
- Y**
- y-intercept, 250
- Z**
- z transform, § 3.4(65), § 7.1(141), § 13.6(227), § 14.2(236)
 - z-plane, 231, § 14.7(254)
 - z-transform, § 14.1(231), 231, § 14.2(236), 237, § 14.5(249)
 - z-transforms, 236
 - zero, § 3.4(65), § 13.4(223), § 13.6(227), § 14.7(254), § 14.8(258)
 - zero-input response, 47
 - zero-state response, 47
 - zeros, 227, 250, 254

Attributions

Collection: *Signals and Systems*

Edited by: Richard Baraniuk

URL: <http://cnx.org/content/col10064/1.11/>

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Signal Classifications and Properties"

By: Melissa Selik, Richard Baraniuk, Michael Haag

URL: <http://cnx.org/content/m10057/2.17/>

Pages: 1-9

Copyright: Melissa Selik, Richard Baraniuk, Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Size of A Signal: Norms"

By: Richard Baraniuk

URL: <http://cnx.org/content/m12363/1.2/>

Pages: 9-19

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Signal Operations"

By: Richard Baraniuk

URL: <http://cnx.org/content/m10125/2.8/>

Pages: 19-22

Copyright: Richard Baraniuk, Adam Blair

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Useful Signals"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10058/2.12/>

Pages: 22-25

Copyright: Melissa Selik, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "The Impulse Function"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10059/2.19/>

Pages: 25-28

Copyright: Melissa Selik, Richard Baraniuk, Adam Blair

License: <http://creativecommons.org/licenses/by/1.0>

Module: "The Complex Exponential"

By: Richard Baraniuk

URL: <http://cnx.org/content/m10060/2.20/>

Pages: 28-31

Copyright: Richard Baraniuk, Adam Blair

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Discrete-Time Signals"

By: Don Johnson

URL: <http://cnx.org/content/m0009/2.23/>

Pages: 31-34

Copyright: Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "System Classifications and Properties"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10084/2.19/>

Pages: 37-41

Copyright: Melissa Selik, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Properties of Systems"

By: Thanos Antoulas, JP Slavinsky

URL: <http://cnx.org/content/m2102/2.16/>

Pages: 41-46

Copyright: Thanos Antoulas, JP Slavinsky

License: <http://creativecommons.org/licenses/by/1.0>

Module: "CT Linear Systems and Differential Equations"

By: Michael Haag, Richard Baraniuk

URL: <http://cnx.org/content/m10855/2.6/>

Pages: 47-52

Copyright: Michael Haag, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Continuous-Time Convolution"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10085/2.26/>

Pages: 53-59

Copyright: Melissa Selik, Richard Baraniuk, Adam Blair

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Properties of Convolution"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10088/2.14/>

Pages: 59-65

Copyright: Melissa Selik, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "BIBO Stability"

By: Richard Baraniuk

URL: <http://cnx.org/content/m10113/2.9/>

Pages: 65-67

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Discrete-Time Systems in the Time-Domain"

By: Don Johnson

URL: <http://cnx.org/content/m10251/2.22/>

Pages: 69-72

Copyright: Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Discrete-Time Convolution"

By: Ricardo Radaelli-Sanchez, Richard Baraniuk

URL: <http://cnx.org/content/m10087/2.18/>

Pages: 73-78

Copyright: Ricardo Radaelli-Sanchez, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Circular Convolution and the DFT"

By: Justin Romberg

URL: <http://cnx.org/content/m10786/2.8/>

Pages: 158-162

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Linear Constant-Coefficient Difference Equations"

By: Richard Baraniuk

URL: <http://cnx.org/content/m12325/1.3/>

Pages: 82-83

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Solving Linear Constant-Coefficient Difference Equations"

By: Richard Baraniuk

URL: <http://cnx.org/content/m12326/1.3/>

Pages: 83-88

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Linear Algebra: The Basics"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10734/2.4/>

Pages: 91-95

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Eigenvectors and Eigenvalues"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10736/2.7/>

Pages: 96-100

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Matrix Diagonalization"

By: Michael Haag

URL: <http://cnx.org/content/m10738/2.5/>

Pages: 101-103

Copyright: Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Eigen-stuff in a Nutshell"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10742/2.4/>

Pages: 104-104

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Eigenfunctions of LTI Systems"

By: Justin Romberg

URL: <http://cnx.org/content/m10500/2.7/>

Pages: 105-107

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Fourier Transform Properties"

By: Don Johnson

URL: <http://cnx.org/content/m0045/2.8/>

Pages: 108-108

Copyright: Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Periodic Signals"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10744/2.6/>

Pages: 111-112

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Fourier Series: Eigenfunction Approach"

By: Justin Romberg

URL: <http://cnx.org/content/m10496/2.21/>

Pages: 112-115

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Derivation of Fourier Coefficients Equation"

By: Michael Haag

URL: <http://cnx.org/content/m10733/2.6/>

Pages: 115-116

Copyright: Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Fourier Series in a Nutshell"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10751/2.3/>

Pages: 116-119

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Fourier Series Properties"

By: Justin Romberg, Benjamin Fite

URL: <http://cnx.org/content/m10740/2.7/>

Pages: 119-122

Copyright: Justin Romberg, Benjamin Fite

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Symmetry Properties of the Fourier Series"

By: Justin Romberg

URL: <http://cnx.org/content/m10838/2.4/>

Pages: 122-126

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Circular Convolution Property of Fourier Series"

By: Justin Romberg

URL: <http://cnx.org/content/m10839/2.4/>

Pages: 126-127

Copyright: Richard Baraniuk, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Fourier Series and LTI Systems"

By: Justin Romberg

URL: <http://cnx.org/content/m10752/2.7/>

Pages: 127-130

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Convergence of Fourier Series"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10745/2.3/>

Pages: 130-132

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Dirichlet Conditions"

By: Ricardo Radaelli-Sanchez

URL: <http://cnx.org/content/m10089/2.9/>

Pages: 132-134

Copyright: Ricardo Radaelli-Sanchez

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Gibbs's Phenomena"

By: Ricardo Radaelli-Sanchez, Richard Baraniuk

URL: <http://cnx.org/content/m10092/2.9/>

Pages: 134-137

Copyright: Ricardo Radaelli-Sanchez, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Fourier Series Wrap-Up"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10749/2.4/>

Pages: 137-138

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Fourier Analysis"

By: Richard Baraniuk

URL: <http://cnx.org/content/m10096/2.10/>

Pages: 141-142

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Fourier Analysis in Complex Spaces"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10784/2.7/>

Pages: 142-149

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Matrix Equation for the DTFS"

By: Roy Ha

URL: <http://cnx.org/content/m10771/2.6/>

Pages: 149-150

Copyright: Roy Ha

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Periodic Extension to DTFS"

By: Roy Ha

URL: <http://cnx.org/content/m10778/2.8/>

Pages: 150-154

Copyright: Roy Ha

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Circular Shifts"

By: Justin Romberg

URL: <http://cnx.org/content/m10780/2.6/>

Pages: 154-158

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Circular Convolution and the DFT"

By: Justin Romberg

URL: <http://cnx.org/content/m10786/2.8/>

Pages: 158-162

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "DFT: Fast Fourier Transform"

By: Don Johnson

URL: <http://cnx.org/content/m0504/2.8/>

Pages: 165-165

Copyright: Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "The Fast Fourier Transform (FFT)"

By: Justin Romberg

URL: <http://cnx.org/content/m10783/2.5/>

Pages: 166-167

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Deriving the Fast Fourier Transform"

By: Don Johnson

URL: <http://cnx.org/content/m0528/2.7/>

Pages: 167-169

Copyright: Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Convergence of Sequences"

By: Richard Baraniuk

URL: <http://cnx.org/content/m10883/2.4/>

Pages: 171-172

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Convergence of Vectors"

By: Michael Haag

URL: <http://cnx.org/content/m10894/2.2/>

Pages: 173-176

Copyright: Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Uniform Convergence of Function Sequences"

By: Michael Haag, Richard Baraniuk

URL: <http://cnx.org/content/m10895/2.5/>

Pages: 176-177

Copyright: Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Discrete Fourier Transformation"

By: Phil Schniter

URL: <http://cnx.org/content/m10421/2.10/>

Pages: 179-180

Copyright: Phil Schniter

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Discrete Fourier Transform (DFT)"

By: Don Johnson

URL: <http://cnx.org/content/m10249/2.26/>

Pages: 181-183

Copyright: Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Table of Common Fourier Transforms"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10099/2.9/>

Pages: 183-184

Copyright: Melissa Selik, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Discrete-Time Fourier Transform (DTFT)"

By: Richard Baraniuk

URL: <http://cnx.org/content/m10108/2.11/>

Pages: 184-184

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Discrete-Time Fourier Transform Properties"

By: Don Johnson

URL: <http://cnx.org/content/m0506/2.6/>

Pages: 185-185

Copyright: Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Discrete-Time Fourier Transform Pair"

By: Don Johnson

URL: <http://cnx.org/content/m0525/2.6/>

Pages: 185-186

Copyright: Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "DTFT Examples"

By: Don Johnson

URL: <http://cnx.org/content/m0524/2.10/>

Pages: 186-189

Copyright: Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Continuous-Time Fourier Transform (CTFT)"

By: Richard Baraniuk, Melissa Selik

URL: <http://cnx.org/content/m10098/2.9/>

Pages: 191-192

Copyright: Richard Baraniuk, Melissa Selik

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Properties of the Continuous-Time Fourier Transform"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10100/2.13/>

Pages: 192-195

Copyright: Melissa Selik, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Sampling"

By: Justin Romberg

URL: <http://cnx.org/content/m10798/2.6/>

Pages: 197-201

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Reconstruction"

By: Justin Romberg

URL: <http://cnx.org/content/m10788/2.5/>

Pages: 201-205

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "More on Perfect Reconstruction"

Used here as: "More on Reconstruction"

By: Roy Ha, Justin Romberg

URL: <http://cnx.org/content/m10790/2.4/>

Pages: 205-207

Copyright: Roy Ha, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Nyquist Theorem"

By: Justin Romberg

URL: <http://cnx.org/content/m10791/2.5/>

Pages: 207-208

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Aliasing"

By: Justin Romberg, Don Johnson

URL: <http://cnx.org/content/m10793/2.6/>

Pages: 209-212

Copyright: Justin Romberg, Don Johnson

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Anti-Aliasing Filters"

By: Justin Romberg

URL: <http://cnx.org/content/m10794/2.4/>

Pages: 212-213

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Discrete Time Processing of Continuous Time Signals"

By: Justin Romberg

URL: <http://cnx.org/content/m10797/2.9/>

Pages: 214-216

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "The Laplace Transforms"

By: Richard Baraniuk

URL: <http://cnx.org/content/m10110/2.12/>

Pages: 219-221

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Properties of the Laplace Transform"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10117/2.9/>

Pages: 222-222

Copyright: Melissa Selik, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Table of Common Laplace Transforms"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10111/2.10/>

Pages: 223-223

Copyright: Melissa Selik, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Region of Convergence for the Laplace Transform"

By: Richard Baraniuk

URL: <http://cnx.org/content/m10114/2.8/>

Pages: 223-225

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "The Inverse Laplace Transform"

By: Steven Cox

URL: <http://cnx.org/content/m10170/2.8/>

Pages: 225-227

Copyright: Steven Cox

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Poles and Zeros"

By: Richard Baraniuk

URL: <http://cnx.org/content/m10112/2.11/>

Pages: 227-229

Copyright: Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "The Z Transform: Definition"

By: Benjamin Fite

URL: <http://cnx.org/content/m10549/2.9/>

Pages: 231-236

Copyright: Benjamin Fite

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Table of Common z-Transforms"

By: Melissa Selik, Richard Baraniuk

URL: <http://cnx.org/content/m10119/2.13/>

Pages: 236-237

Copyright: Melissa Selik, Richard Baraniuk

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Region of Convergence for the Z-transform"

By: Benjamin Fite

URL: <http://cnx.org/content/m10622/2.5/>

Pages: 237-246

Copyright: Benjamin Fite

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Inverse Z-Transform"

By: Benjamin Fite

URL: <http://cnx.org/content/m10651/2.4/>

Pages: 246-248

Copyright: Benjamin Fite

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Rational Functions"

By: Michael Haag

URL: <http://cnx.org/content/m10593/2.7/>

Pages: 249-251

Copyright: Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Difference Equation"

By: Michael Haag

URL: <http://cnx.org/content/m10595/2.5/>

Pages: 251-254

Copyright: Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Understanding Pole/Zero Plots on the Z-Plane"

By: Michael Haag

URL: <http://cnx.org/content/m10556/2.8/>

Pages: 254-258

Copyright: Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Filter Design using the Pole/Zero Plot of a Z-Transform"

By: Michael Haag

URL: <http://cnx.org/content/m10548/2.9/>

Pages: 258-261

Copyright: Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Vector Spaces"

By: Michael Haag, Steven Cox, Justin Romberg

URL: <http://cnx.org/content/m10767/2.4/>

Pages: 263-264

Copyright: Michael Haag, Steven Cox, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Norms"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10768/2.4/>

Pages: 265-267

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Inner Products"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10755/2.6/>

Pages: 268-269

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Hilbert Spaces"

By: Justin Romberg

URL: <http://cnx.org/content/m10840/2.4/>

Pages: 270-270

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Cauchy-Schwarz Inequality"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10757/2.5/>

Pages: 270-277

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Common Hilbert Spaces"

By: Roy Ha, Justin Romberg

URL: <http://cnx.org/content/m10759/2.5/>

Pages: 277-280

Copyright: Roy Ha, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Types of Basis"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10772/2.5/>

Pages: 280-283

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Orthonormal Basis Expansions"

By: Michael Haag, Justin Romberg

URL: <http://cnx.org/content/m10760/2.4/>

Pages: 283-287

Copyright: Michael Haag, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Function Space"

By: Justin Romberg

URL: <http://cnx.org/content/m10770/2.4/>

Pages: 287-288

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Haar Wavelet Basis"

By: Roy Ha, Justin Romberg

URL: <http://cnx.org/content/m10764/2.7/>

Pages: 288-295

Copyright: Roy Ha, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Orthonormal Bases in Real and Complex Spaces"

By: Justin Romberg

URL: <http://cnx.org/content/m10765/2.7/>

Pages: 295-297

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Plancharel and Parseval's Theorems"

By: Justin Romberg

URL: <http://cnx.org/content/m10769/2.5/>

Pages: 297-298

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Approximation and Projections in Hilbert Space"

By: Justin Romberg

URL: <http://cnx.org/content/m10766/2.7/>

Pages: 298-300

Copyright: Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Homework 1"

By: Richard Baraniuk, Justin Romberg

URL: <http://cnx.org/content/m10826/2.9/>

Pages: 303-307

Copyright: Richard Baraniuk, Justin Romberg

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Homework 1 Solutions"

By: Justin Romberg, Richard Baraniuk, Michael Haag

URL: <http://cnx.org/content/m10830/2.4/>

Pages: 307-318

Copyright: Justin Romberg, Richard Baraniuk, Michael Haag

License: <http://creativecommons.org/licenses/by/1.0>

Module: "Viewing Embedded LabVIEW Content"

By: Matthew Hutchinson

URL: <http://cnx.org/content/m13753/1.3/>

Pages: 319-319

Copyright: Matthew Hutchinson

License: <http://creativecommons.org/licenses/by/2.0/>

Signals and Systems

This course deals with signals, systems, and transforms, from their theoretical mathematical foundations to practical implementation in circuits and computer algorithms. At the conclusion of ELEC 301, you should have a deep understanding of the mathematics and practical issues of signals in continuous and discrete time, linear time invariant systems, convolution, and Fourier transforms.

About Connexions

Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.