SIGNALS AND SYSTEMS



Richard Baraniuk et al.
Rice University



Signals and Systems

Richard Baraniuk et al.

Rice University

This text is disseminated via the Open Education Resource (OER) LibreTexts Project (https://LibreTexts.org) and like the thousands of other texts available within this powerful platform, it is freely available for reading, printing, and "consuming."

The LibreTexts mission is to bring together students, faculty, and scholars in a collaborative effort to provide an accessible, and comprehensive platform that empowers our community to develop, curate, adapt, and adopt openly licensed resources and technologies; through these efforts we can reduce the financial burden born from traditional educational resource costs, ensuring education is more accessible for students and communities worldwide.

Most, but not all, pages in the library have licenses that may allow individuals to make changes, save, and print this book. Carefully consult the applicable license(s) before pursuing such effects. Instructors can adopt existing LibreTexts texts or Remix them to quickly build course-specific resources to meet the needs of their students. Unlike traditional textbooks, LibreTexts' web based origins allow powerful integration of advanced features and new technologies to support learning.



LibreTexts is the adaptable, user-friendly non-profit open education resource platform that educators trust for creating, customizing, and sharing accessible, interactive textbooks, adaptive homework, and ancillary materials. We collaborate with individuals and organizations to champion open education initiatives, support institutional publishing programs, drive curriculum development projects, and more.

The LibreTexts libraries are Powered by NICE CXone Expert and was supported by the Department of Education Open Textbook Pilot Project, the California Education Learning Lab, the UC Davis Office of the Provost, the UC Davis Library, the California State University Affordable Learning Solutions Program, and Merlot. This material is based upon work supported by the National Science Foundation under Grant No. 1246120, 1525057, and 1413739.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation nor the US Department of Education.

Have questions or comments? For information about adoptions or adaptions contact info@LibreTexts.org or visit our main website at https://LibreTexts.org.

This text was compiled on 05/13/2026

TABLE OF CONTENTS

Licensing

About the Book

1: Introduction to Signals

- 1.1: Signal Classifications and Properties
- 1.2: Signal Size and Norms
- 1.3: Signal Operations
- 1.4: Common Continuous Time Signals
- 1.5: Common Discrete Time Signals
- 1.6: Continuous Time Impulse Function
- 1.7: Discrete Time Impulse Function
- 1.8: Continuous Time Complex Exponential
- 1.9: Discrete Time Complex Exponential

2: Introduction to Systems

- 2.1: System Classifications and Properties
- 2.2: Linear Time Invariant Systems

3: Time Domain Analysis of Continuous Time Systems

- 3.1: Continuous Time Systems
- 3.2: Continuous Time Impulse Response
- 3.3: Continuous Time Convolution
- 3.4: Properties of Continuous Time Convolution
- 3.5: Eigenfunctions of Continuous Time LTI Systems
- 3.6: BIBO Stability of Continuous Time Systems
- 3.7: Linear Constant Coefficient Differential Equations
- 3.8: Solving Linear Constant Coefficient Differential Equations

4: Time Domain Analysis of Discrete Time Systems

- 4.1: Discrete Time Systems
- 4.2: Discrete Time Impulse Response
- 4.3: Discrete Time Convolution
- 4.4: Properties of Discrete Time Convolution
- 4.5: Eigenfunctions of Discrete Time LTI Systems
- 4.6: BIBO Stability of Discrete Time Systems
- 4.7: Linear Constant Coefficient Difference Equations
- 4.8: Solving Linear Constant Coefficient Difference Equations

5: Introduction to Fourier Analysis

- 5.1: Introduction to Fourier Analysis

6: Continuous Time Fourier Series (CTFS)

- 6.1: Continuous Time Periodic Signals
- 6.2: Continuous Time Fourier Series (CTFS)
- 6.3: Common Fourier Series
- 6.4: Properties of the CTFS
- 6.5: Continuous Time Circular Convolution and the CTFS
- 6.6: Convergence of Fourier Series
- 6.7: Gibbs Phenomena

7: Discrete Time Fourier Series (DTFS)

- 7.1: Discrete Time Periodic Signals
- 7.2: Discrete Time Fourier Series (DTFS)
- 7.3: Common Discrete Fourier Series
- 7.4: Properties of the DTFS
- 7.5: Discrete Time Circular Convolution and the DTFS

8: Continuous Time Fourier Transform (CTFT)

- 8.1: Continuous Time Aperiodic Signals
- 8.2: Continuous Time Fourier Transform (CTFT)
- 8.3: Common Fourier Transforms
- 8.4: Properties of the CTFT
- 8.5: Continuous Time Convolution and the CTFT

9: Discrete Time Fourier Transform (DTFT)

- 9.1: Discrete Time Aperiodic Signals
- 9.2: Discrete Time Fourier Transform (DTFT)
- 9.3: Common Discrete Time Fourier Transforms
- 9.4: Properties of the DTFT
- 9.5: Discrete Time Convolution and the DTFT

10: Sampling and Reconstruction

- 10.1: Signal Sampling
- 10.2: Sampling Theorem
- 10.3: Signal Reconstruction
- 10.4: Perfect Reconstruction
- 10.5: Aliasing Phenomena
- 10.6: Anti-Aliasing Filters
- 10.7: Discrete Time Processing of Continuous Time Signals

11: Laplace Transform and Continuous Time System Design

- 11.1: Laplace Transform
- 11.2: Common Laplace Transforms
- 11.3: Properties of the Laplace Transform
- 11.4: Inverse Laplace Transform
- 11.5: Poles and Zeros in the S-Plane
- 11.6: Region of Convergence for the Laplace Transform
- 11.7: Rational Functions and the Laplace Transform

- 11.8: Differential Equations
- 11.9: Continuous Time Filter Design

12: Z-Transform and Discrete Time System Design

- 12.1: Z-Transform
- 12.2: Common Z-Transforms
- 12.3: Properties of the Z-Transform
- 12.4: Inverse Z-Transform
- 12.5: Poles and Zeros in the Z-Plane
- 12.6: Region of Convergence for the Z-Transform
- 12.7: Rational Functions and the Z-Transform
- 12.8: Difference Equations
- 12.9: Discrete Time Filter Design

13: Capstone Signal Processing Topics

- 13.1: DFT- Fast Fourier Transform
- 13.2: The Fast Fourier Transform (FFT)
- 13.3: Deriving the Fast Fourier Transform
- 13.4: Matched Filter Detector

14: Appendix A- Linear Algebra Overview

- 14.1: Basic Linear Algebra
- 14.2: Eigenvectors and Eigenvalues
- 14.3: Matrix Diagonalization
- 14.4: Eigen-Stuff in a Nutshell
- 14.5: Eigenfunctions of LTI Systems

15: Appendix B- Hilbert Spaces Overview

- 15.1: Fields and Complex Numbers
- 15.2: Vector Spaces
- 15.3: Norms
- 15.4: Inner Products
- 15.5: Hilbert Spaces
- 15.6: Cauchy-Schwarz Inequality
- 15.7: Common Hilbert Spaces
- 15.8: Types of Bases
- 15.9: Orthonormal Basis Expansions
- 15.10: Function Space
- 15.11: Haar Wavelet Basis
- 15.12: Orthonormal Bases in Real and Complex Spaces
- 15.13: Plancharel and Parseval's Theorems
- 15.14: Approximation and Projections in Hilbert Space

16: Appendix C- Analysis Topics Overview

- 16.1: Introduction to Metric Spaces
- 16.2: Convergence of Sequences
- 16.3: Convergence of Sequences of Vectors
- 16.4: Uniform Convergence of Function Sequences

17: Appendix D- Viewing Interactive Content

- 17.1: Viewing Embedded LabVIEW Content in Connexions
- 17.2: Getting Started with Mathematica

Index

Glossary

Detailed Licensing

Licensing

*A detailed breakdown of this resource's licensing can be found in **Back Matter/Detailed Licensing**.*

About the Book

This textbook offers a comprehensive introduction to the fundamental principles of **signals and systems**, with balanced coverage of both **continuous-time** and **discrete-time** domains. It begins with the basic **classification and operations on signals** (Chapter 1) and **systems** (Chapter 2), laying a solid foundation for understanding linear, time-invariant behavior.

Chapters 3 and 4 provide detailed **time-domain analysis** of systems, including **impulse responses, convolution**, and **difference/differential equations**, with an emphasis on **BIBO stability** and **eigenfunctions**.

The core of the text (Chapters 5–9) is dedicated to **Fourier analysis**, beginning with periodic signals (Fourier Series) and extending to aperiodic signals (Fourier Transforms) in both continuous and discrete time. These chapters also explore the **properties of transforms**, **circular convolution**, and convergence issues such as **Gibbs phenomena**.

Chapter 10 introduces **sampling and signal reconstruction**, including the **sampling theorem, aliasing**, and **anti-aliasing filters**, and explains how discrete-time processing can be applied to continuous-time signals.

Chapters 11 and 12 delve into system analysis and design using the **Laplace Transform** and **Z-Transform**, covering **pole-zero analysis**, **regions of convergence**, and **filter design** techniques.

Chapter 13 serves as a capstone, introducing **modern signal processing tools** such as the **Fast Fourier Transform (FFT)** and **matched filtering** for detection.

The textbook concludes with **four appendices**:

- **Appendix A** covers key **linear algebra** concepts, especially eigenvalue problems and diagonalization.
- **Appendix B** provides a foundation in **Hilbert spaces**, norms, inner products, orthonormal bases, and function space theory.
- **Appendix C** introduces essential **analysis topics**, including convergence in metric spaces and uniform convergence of function sequences.
- **Appendix D** gives instructions for **interactive content viewing**, including LabVIEW and Mathematica integration.

CHAPTER OVERVIEW

1: Introduction to Signals

1.1: Signal Classifications and Properties
1.2: Signal Size and Norms
1.3: Signal Operations
1.4: Common Continuous Time Signals
1.5: Common Discrete Time Signals
1.6: Continuous Time Impulse Function
1.7: Discrete Time Impulse Function
1.8: Continuous Time Complex Exponential
1.9: Discrete Time Complex Exponential

This page titled 1: Introduction to Signals is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

1.1: Signal Classifications and Properties

Introduction

This module will begin our study of signals and systems by laying out some of the fundamentals of signal classification. It is essentially an introduction to the important definitions and properties that are fundamental to the discussion of signals and systems, with a brief discussion of each.

Classifications of Signals

Continuous-Time vs. Discrete-Time

As the names suggest, this classification is determined by whether or not the time axis is **discrete** (countable) or **continuous** (Figure 1.1.1). A continuous-time signal will contain a value for all real numbers along the time axis. In contrast to this, a discrete-time signal, often created by sampling a continuous signal, will only have values at equally spaced intervals along the time axis.

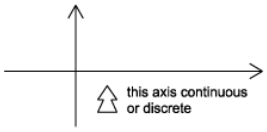


Figure 1.1.1

Analog vs. Digital

The difference between **analog** and **digital** is similar to the difference between continuous-time and discrete-time. However, in this case the difference involves the values of the function. Analog corresponds to a continuous set of possible function values, while digital corresponds to a discrete set of possible function values. An common example of a digital signal is a binary sequence, where the values of the function can only be one or zero.

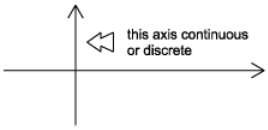


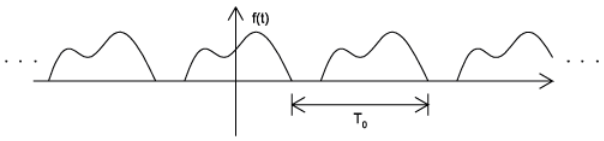
Figure 1.1.2

Periodic vs. Aperiodic

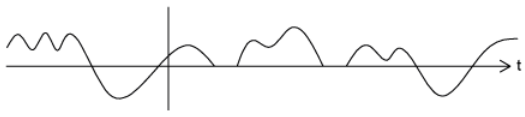
Periodic signals repeat with some **period** T, while aperiodic, or nonperiodic, signals do not (Figure 1.1.3). We can define a periodic function through the following mathematical expression, where t can be any number and T is a positive constant:

$$f(t) = f(t+T) \tag{1.1.1}$$

fundamental period of our function, $f(t)$, is the smallest value of T that the still allows Equation 1.1.1 to be true.



(a)



(b)

Figure 1.1.3: (a) A periodic signal with period T_0 (b) An aperiodic signal

Finite vs. Infinite Length

Another way of classifying a signal is in terms of its length along its time axis. Is the signal defined for all possible values of time, or for only certain values of time? Mathematically speaking, $f(t)$ is a **finite-length signal** if it is **defined** only over a finite interval

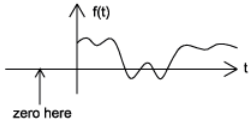
$$t_1 < t < t_2$$

where $t_1 < t_2$. Similarly, an **infinite-length signal**, $f(t)$, is defined for all values:

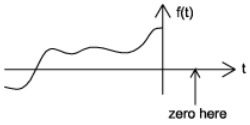
$$-\infty < t < \infty$$

Causal vs. Anticausal vs. Noncausal

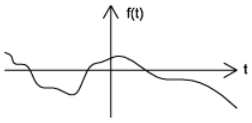
Causal signals are signals that are zero for all negative time, while **anticausal** are signals that are zero for all positive time. **Noncausal** signals are signals that have nonzero values in both positive and negative time (Figure 1.1.4).



(a)



(b)

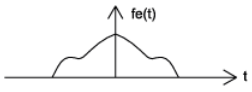


(c)

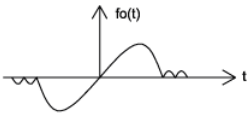
Figure 1.1.4: (a) A causal signal (b) An anticausal signal (c) A noncausal signal

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 1.1.5).



(a)



(b)

Figure 1.1.5: (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.

$$f(t) = \frac{1}{2}(f(t) + f(-t)) + \frac{1}{2}(f(t) - f(-t)) \tag{1.1.2}$$

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 1.1.6).

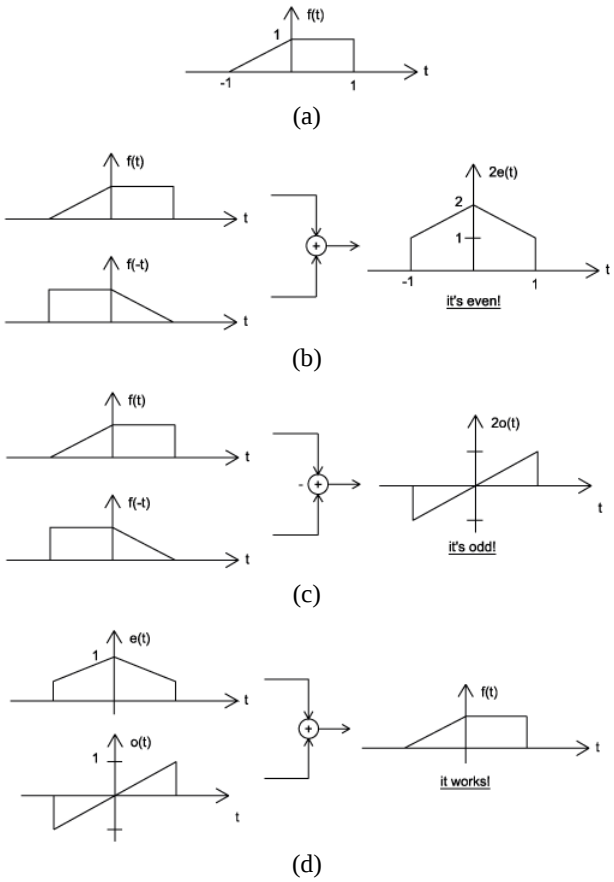


Figure 1.1.6: (a) The signal we will decompose using odd-even decomposition (b) Even part: $e(t) = \frac{1}{2}(f(t) + f(-t))$ (c) Odd part: $o(t) = \frac{1}{2}(f(t) - f(-t))$ (d) Check: $e(t) + o(t) = f(t)$

Deterministic vs. Random

A **deterministic signal** is a signal in which each value of the signal is fixed, being determined by a mathematical expression, rule, or table. On the other hand, the values of a random signal are not strictly defined, but are subject to some amount of variability.

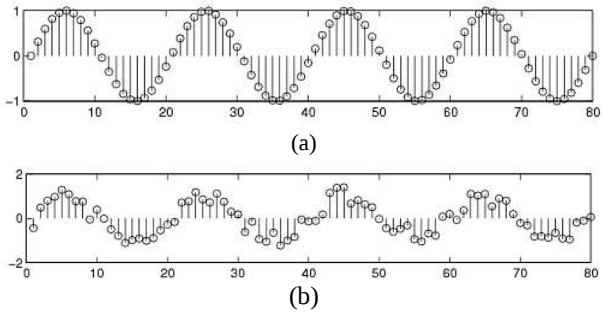


Figure 1.1.7: (a) Deterministic signal (b) Random signal

Consider the signal defined for all real t described by

$$f(t) = \begin{cases} \sin(2\pi t)/t & t \geq 1 \\ 0 & t < 1 \end{cases}$$

This signal is continuous time, analog, aperiodic, infinite length, causal, neither even nor odd, and, by definition, deterministic.

Signal Classifications Summary

This module describes just some of the many ways in which signals can be classified. They can be continuous time or discrete time, analog or digital, periodic or aperiodic, finite or infinite, and deterministic or random. We can also divide them based on their causality and symmetry properties.

This page titled 1.1: Signal Classifications and Properties is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

1.2: Signal Size and Norms

Introduction

The "size" of a signal would involve some notion of its strength. We use the mathematical concept of the norm to quantify this concept for both continuous-time and discrete-time signals. As there are several types of norms that can be defined for signals, there are several different conceptions of signal size.

Signal Energy

Infinite Length, Continuous Time Signals

The most commonly encountered notion of the energy of a signal defined on $\mathbb{R}$ is the L_2 norm defined by the square root of the integral of the square of the signal, for which the notation

$$\|f\|_2 = \left(\int_{-\infty}^{\infty} |f(t)|^2 dt \right)^{1/2}. \tag{1.2.1}$$

However, this idea can be generalized through definition of the L_p norm, which is given by

$$\|f\|_p = \left(\int_{-\infty}^{\infty} |f(t)|^p dt \right)^{1/p} \tag{1.2.2}$$

for all $1 \leq p < \infty$. Because of the behavior of this expression as p approaches ∞, we furthermore define

$$\|f\|_\infty = \sup_{t \in \mathbb{R}} [f(t)|, \tag{1.2.3}$$

which is the least upper bound of $|f(t)|$. A signal f is said to belong to the vector space $L_p(\mathbb{R})$ if $\|f\|_p < \infty$.

Example 1.2.1

For example, consider the function defined by

$$f(t) = \begin{cases} 1/t & 1 \leq t \\ 0 & \text{otherwise} \end{cases} \tag{1.2.4}$$

The L_1 norm is

$$\|f\|_1 = \int_{-\infty}^{\infty} |f(t)| dt = \int_{-\infty}^{\infty} \frac{1}{t} dt = \infty.$$

The L_2 norm is

$$\|f\|_2 = \left(\int_{-\infty}^{\infty} |f(t)|^2 dt \right)^{1/2} = \left(\int_{-\infty}^{\infty} \frac{1}{t^2} dt \right)^{1/2} = 1.$$

The L_∞ norm is

$$\|f\|_\infty = \sup_{t \in \mathbb{R}} |f(t)| = \sup_{t \in \mathbb{R}[1,\infty)} \frac{1}{t} = 1.$$

Finite Length, Continuous Time Signals

The most commonly encountered notion of the energy of a signal defined on $\mathbb{R}[a, b]$ is the L_2 norm defined by the square root of the integral of the square of the signal, for which the notation

$$\|f\|_2 = \left(\int_{a}^{b} |f(t)|^2 dt \right)^{1/2}. \tag{1.2.5}$$

However, this idea can be generalized through definition of the L_p norm, which is given by

$$\|f\|_p = \left(\int_a^b |f(t)|^p dt \right)^{1/p} \tag{1.2.6}$$

for all $1 \leq p < \infty$. Because of the behavior of this expression as p approaches ∞, we furthermore define

$$\|f\|_\infty = \sup_{t \in \mathbb{R}[a,b]} |f(t)| \tag{1.2.7}$$

which is the least upper bound of $|f(t)|$. A signal f is said to belong to the vector space $L_p(\mathbb{R}[a,b])$ if $\|f\|_p < \infty$. The periodic extension of such a signal would have infinite energy but finite power.

Example 1.2.2

For example, consider the function defined on $\mathbb{R}[-5,3]$ by

$$f(t) = \begin{cases} t & -5 < t < 3 \\ 0 & \text{otherwise} \end{cases} . \tag{1.2.8}$$

The L_1 norm is

$$\|f\|_1 = \int_{-5}^{3} |f(t)| dt = \int_{-5}^{3} |t| dt = 17.$$

The L_2 norm is

$$\|f\|_2 = \left(\int_{-5}^{3} |f(t)|^2 dt \right)^{1/2} = \left(\int_{-5}^{3} |t|^2 dt \right)^{1/2} \approx 7.12$$

The L_∞ norm is

$$\|f\|_\infty = \sup_{t \in \mathbb{R}[-5,3]} |t| = 5.$$

Infinite Length, Discrete Time Signals

The most commonly encountered notion of the energy of a signal defined on $\mathbb{Z}$ is the l_2 norm defined by the square root of the sumation of the square of the signal, for which the notation

$$\|x[n]\|_2 = \left(\sum_{n=-\infty}^{\infty} |x[n]|^2 \right)^{1/2} . \tag{1.2.9}$$

However, this idea can be generalized through definition of the l_p norm, which is given by

$$\|x[n]\|_p = \left(\sum_{n=-\infty}^{\infty} |x[n]|^p \right)^{1/p} \tag{1.2.10}$$

for all $1 \leq p < \infty$. Because of the behavior of this expression as p approaches ∞, we furthermore define

$$\|x[n]\|_\infty = \sup_{n \in \mathbb{Z}} |x[n]|, \tag{1.2.11}$$

which is the least upper bound of $|x[n]|$. A signal x is said to belong to the vector space $l_p(\mathbb{Z})$ if $\|x[n]\|_p < \infty$.

Example 1.2.3

For example, consider the function defined by

$$x[n] = \begin{cases} 1/n & 1 \leq n \\ 0 & \text{otherwise} \end{cases} . \tag{1.2.12}$$

The l_1 norm is

$$\|x[n]\|_1 = \sum n = -\infty^{\infty} |x[n]| = \sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{1.2.13}$$

The l_2 norm is

$$\|x[n]\|_2 = \left(\sum_{n=-\infty}^{\infty} |x[n]|^2 \right)^{1/2} = \left(\sum_{n=1}^{\infty} \frac{1}{n^2} \right)^{1/2} = \frac{\pi\sqrt{6}}{6} \tag{1.2.14}$$

The l_∞ norm is

$$\|x[n]\|_\infty = \sup_{n \in \mathbb{Z}} |x[n]| = \sup_{n \in \mathbb{Z}[1,\infty)} \frac{1}{n} = 1. \tag{1.2.15}$$

Finite Length, Discrete Time Signals

The most commonly encountered notion of the energy of a signal defined on $\mathbb{Z}[a, b]$ is the l_2 norm defined by the square root of the sumation of the square of the signal, for which the notation

$$\|x[n]\|_2 = \left(\sum_{n=a}^{b} |x[n]|^2 \right)^{1/2}. \tag{1.2.16}$$

However, this idea can be generalized through definition of the l_p norm, which is given by

$$\|x[n]\|_p = \left(\sum_{n=a}^{b} |x[n]|^p \right)^{1/p} \tag{1.2.17}$$

for all $1 \leq p < \infty$. Because of the behavior of this expression as p approaches ∞, we furthermore define

$$\|x[n]\|_\infty = \sup_{n \in \mathbb{Z}[a,b]} |x[n]|, \tag{1.2.18}$$

which is the least upper bound of $|x[n]|$. In this case, this least upper bound is simply the maximum value of $|x[n]|$. A signal $x[n]$ is said to belong to the vector space $l_p(\mathbb{Z}[a, b])$ if $\|x[n]\|_p < \infty$. The periodic extension of such a signal would have infinite energy but finite power.

Example 1.2.4

For example, consider the function defined on $\mathbb{Z}[-5, 3]$ by

$$x[n] = \begin{cases} n & -5 < n < 3 \\ 0 & \text{otherwise} \end{cases}. \tag{1.2.19}$$

The l_1 norm is

$$\|x[n]\|_1 = \sum_{n=-5}^{3} |x[n]| = \sum -5^3 |n| = 21. \tag{1.2.20}$$

The l_2 norm is

$$\|x[n]\|_2 = \left(\sum_{-5}^{3} |x[n]|^2 \right)^{1/2} = \left(\sum_{-5}^{3} |n|^2 dt \right)^{1/2} \approx 8.31 \tag{1.2.21}$$

The l_∞ norm is

$$\|x[n]\|_\infty = \sup_{n \in \mathbb{Z}[-5,3]} |x[n]| = 5. \tag{1.2.22}$$

Signal Norms Summary

The notion of signal size or energy is formally addressed through the mathematical concept of norms. There are many types of norms that can be defined for signals, some of the most important of which have been discussed here. For each type norm and each type of signal domain (continuous or discrete, and finite or infinite) there are vector spaces defined for signals of finite norm. Finally, while nonzero periodic signals have infinite energy, they have finite power if their single period units have finite energy.

This page titled 1.2: Signal Size and Norms is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

1.3: Signal Operations

Introduction

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

Manipulating the Time Parameter

Time Shifting

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

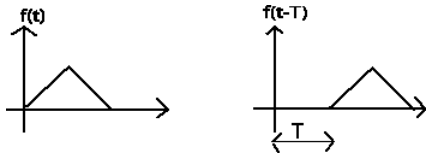


Figure 1.3.1: $f(t-T)$ moves (delays) f to the right by T.

Time Scaling

Time scaling compresses or dilates a signal by multiplying the time variable by some quantity. If that quantity is greater than one, the signal becomes narrower and the operation is called compression, while if the quantity is less than one, the signal becomes wider and is called dilation.

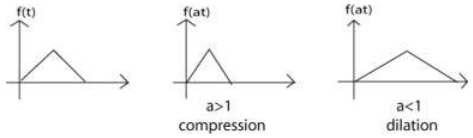


Figure 1.3.2: $f(at)$ compresses f by a.

Example 1.3.1

Given $f(t)$ we would like to plot $f(at-b)$.

Solution

The figure below describes a method to accomplish this.

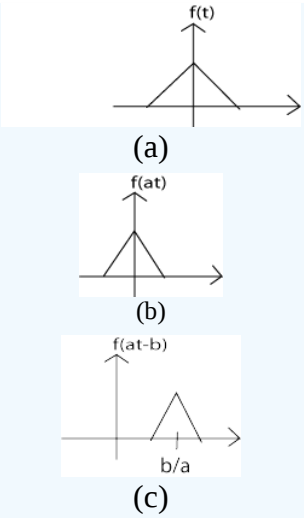


Figure 1.3.3: (a) Begin with $f(t)$ (b) Then replace t with at to get $f(at)$ (c) Finally, replace t with $t-\frac{b}{a}$ to get $f(a(t-\frac{b}{a}))=f(at-b)$

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.

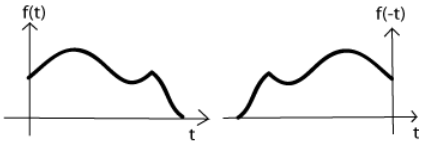


Figure 1.3.4: Reverse the time axis

Time Scaling and Shifting Demonstration

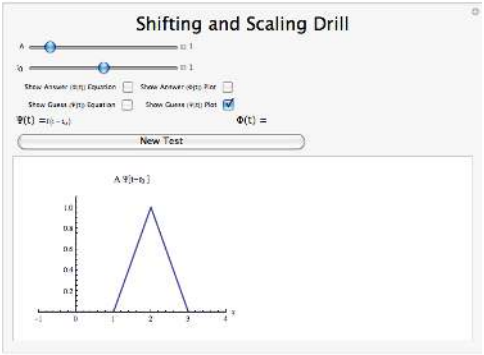


Figure 1.3.5: Download or Interact (when online) with a Mathematica CDF demonstrating Discrete Harmonic Sinusoids.

Signal Operations Summary

Some common operations on signals affect the time parameter of the signal. One of these is time shifting in which a quantity is added to the time parameter in order to advance or delay the signal. Another is the time scaling in which the time parameter is multiplied by a quantity in order to dilate or compress the signal in time. In the event that the quantity involved in the latter operation is negative, time reversal occurs.

This page titled 1.3: Signal Operations is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

1.4: Common Continuous Time Signals

Introduction

Before looking at this module, hopefully you have an idea of what a signal is and what basic classifications and properties a signal can have. In review, a signal is a function defined with respect to an independent variable. This variable is often time but could represent any number of things. Mathematically, continuous time analog signals have continuous independent and dependent variables. This module will describe some useful continuous time analog signals.

Important Continuous Time Signals

Sinusoids

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

$$A\cos(\omega t+\varphi) \tag{1.4.1}$$

where A is the amplitude, ω is the frequency, and φ is the phase. Thus, the period of the sinusoid is

$$T=\frac{2\pi}{\omega} \tag{1.4.2}$$

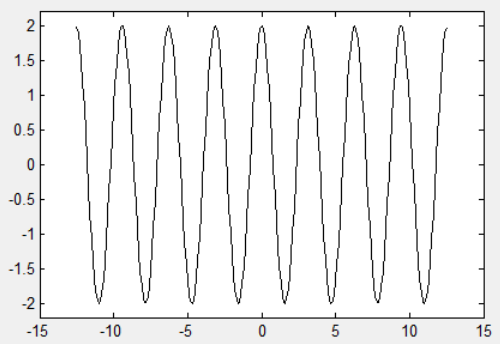


Figure 1.4.1: Sinusoid with $A=2$, $w=2$, and $\varphi=0$.

Complex Exponentials

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

$$Ae^{st}$$

where $s=\sigma+j\omega$ is a complex number in terms of σ, the attenuation constant, and ω the angular frequency.

Unit Impulses

The unit impulse function, also known as the Dirac delta function, is a signal that has infinite height and infinitesimal width. However, because of the way it is defined, it integrates to one. While this signal is useful for the understanding of many concepts, a formal understanding of its definition more involved. The unit impulse is commonly denoted $\delta(t)\delta$ t.

More detail is provided in the section on the continuous time impulse function. For now, it suffices to say that this signal is crucially important in the study of continuous signals, as it allows the sifting property to be used in signal representation and signal decomposition.

Unit Step

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

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

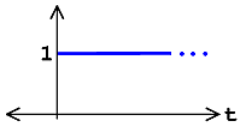


Figure 1.4.2: Continuous-Time Unit-Step Function

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.

Common Continuous Time Signals Summary

Some of the most important and most frequently encountered signals have been discussed in this module. There are, of course, many other signals of significant consequence not discussed here. As you will see later, many of the other more complicated signals will be studied in terms of those listed here. Especially take note of the complex exponentials and unit impulse functions, which will be the key focus of several topics included in this course.

This page titled 1.4: Common Continuous Time Signals is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

1.5: Common Discrete Time Signals

Introduction

Before looking at this module, hopefully you have an idea of what a signal is and what basic classifications and properties a signal can have. In review, a signal is a function defined with respect to an independent variable. This variable is often time but could represent any number of things. Mathematically, discrete time analog signals have discrete independent variables and continuous dependent variables. This module will describe some useful discrete time analog signals.

Important Discrete Time Signals

Sinusoids

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

$$A\cos(\omega n + \varphi)$$

where A is the amplitude, ω is the frequency, and φ is the phase. Because n only takes integer values, the resulting function is only periodic if $\frac{2\pi}{\omega}$ is a rational number.

Discrete-Time Cosine Signal

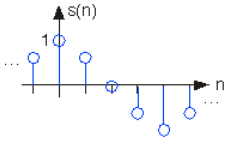


Figure 1.5.1: A discrete-time cosine signal is plotted as a stem plot.

Note that the equation representation for a discrete time sinusoid waveform is not unique.

Complex Exponentials

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

$$z^n$$

where z, is a complex number. The set of complex exponentials for which $|z| = 1$ are a special class, expressed as $e^{j\omega n}$, (where ω is the angular position on the unit circle, in radians).

The discrete time complex exponentials have the following property.

$$e^{j\omega n} = e^{j(\omega + 2\pi)n}$$

Given this property, if we have a complex exponential with frequency $\omega + 2\pi$, then this signal "aliases" to a complex exponential with frequency ω, implying that the equation representations of discrete complex exponentials are not unique.

Unit Impulses

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

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

Unit Sample

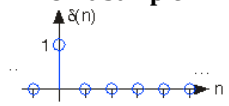


Figure 1.5.2: The unit sample.

More detail is provided in the section on the discrete time impulse function. For now, it suffices to say that this signal is crucially important in the study of discrete signals, as it allows the sifting property to be used in signal representation and signal

decomposition.

Unit Step

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

$$u[n] = \begin{cases} 0 & \text{if } n < 0 \\ 1 & \text{if } n \geq 0 \end{cases}$$

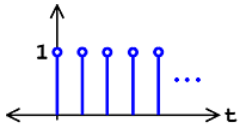


Figure 1.5.3: Discrete-Time Unit-Step Function

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.

Common Discrete Time Signals Summary

Some of the most important and most frequently encountered signals have been discussed in this module. There are, of course, many other signals of significant consequence not discussed here. As you will see later, many of the other more complicated signals will be studied in terms of those listed here. Especially take note of the complex exponentials and unit impulse functions, which will be the key focus of several topics included in this course.

This page titled 1.5: Common Discrete Time Signals is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

1.6: Continuous Time Impulse Function

Introduction

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 some other 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, in this systems and signals course.

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 in continuous-time. Informally, this function is one that is infinitesimally narrow, infinitely tall, yet integrates to one. Perhaps the simplest way to visualize this is as a rectangular pulse from $a - \frac{\varepsilon}{2}$ to $a + \frac{\varepsilon}{2}$ with a height of $\frac{1}{\varepsilon}$. As we take the limit of this setup as ε approaches 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 $\delta(t)$.

$$\int_{-\infty}^{\infty} \delta(t)\mathrm{d}t = 1$$

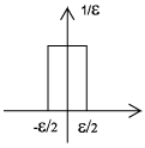


Figure 1.6.1: This is one way to visualize the Dirac Delta Function.

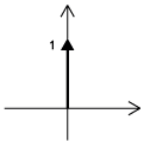


Figure 1.6.2: Since it is quite difficult to draw something that is infinitely tall, we represent the Dirac with an arrow centered at the point it is applied. If we wish to scale it, we may write the value it is scaled by next to the point of the arrow. This is a unit impulse (no scaling).

Below is a brief list a few important properties of the unit impulse without going into detail of their proofs.

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

The last of these is especially important as it gives rise to the sifting property of the dirac delta function, which selects the value of a function at a specific time and is especially important in studying the relationship of an operation called convolution to time domain analysis of linear time invariant systems. The sifting property is shown and derived below.

$$\int_{-\infty}^{\infty} f(t)\delta(t)dt = \int_{-\infty}^{\infty} f(0)\delta(t)dt = f(0)\int_{-\infty}^{\infty} \delta(t)dt = f(0)$$

Unit Impulse Limiting Demonstration

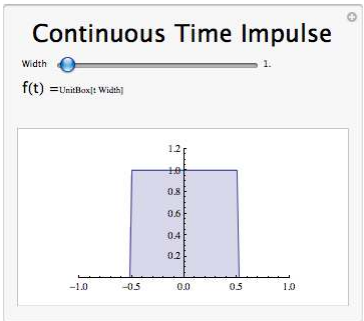


Figure 1.6.3: Click on the above thumbnail image (when online) to download an interactive Mathematica Player demonstrating the Continuous Time Impulse Function.

Continuous Time Unit Impulse Summary

The continuous time unit impulse function, also known as the Dirac delta function, is of great importance to the study of signals and systems. Informally, it is a function with infinite height ant infinitesimal width that integrates to one, which can be viewed as the limiting behavior of a unit area rectangle as it narrows while preserving area. It has several important properties that will appear again when studying systems.

This page titled 1.6: Continuous Time Impulse Function is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

1.7: Discrete Time Impulse Function

Introduction

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 some other 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, in this systems and signals course.

Unit Sample Function

The **unit sample function**, often referred to as the unit impulse or delta function, is the function that defines the idea of a unit impulse in discrete time. There are not nearly as many intricacies involved in its definition as there are in the definition of the Dirac delta function, the continuous time impulse function. The unit sample function simply takes a value of one at $n = 0$ and a value of zero elsewhere. The impulse function is often written as $\delta[n]$.

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

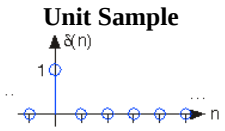


Figure 1.7.1: The unit sample.

Below we will briefly list a few important properties of the unit impulse without going into detail of their proofs.

Unit Impulse Properties

- $\delta[n] = \delta[-n]$
- $\delta[n] = u[n] - u[n-1]$
- $x[n]\delta[n] = x[0]\delta[n]$

$$\sum_{n=-\infty}^{\infty} x[n]\delta[n] = \sum_{n=-\infty}^{\infty} x[0]\delta[n] = x[0] \sum_{n=-\infty}^{\infty} \delta[n] = x[0]$$

Discrete Time Impulse Response Demonstration

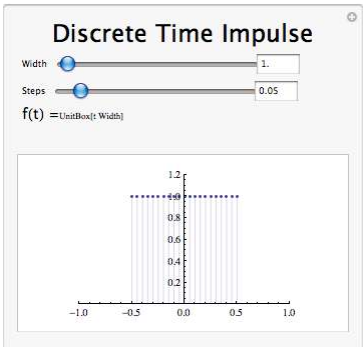


Figure 1.7.2: Interact(when online) with a Mathematica CDF demonstrating the Discrete Time Impulse Function.

Discrete Time Unit Impulse Summary

The discrete time unit impulse function, also known as the unit sample function, is of great importance to the study of signals and systems. The function takes a value of one at time $n = 0$ and a value of zero elsewhere. It has several important properties that will appear again when studying systems.

This page titled 1.7: Discrete Time Impulse Function is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

1.8: Continuous Time Complex Exponential

Introduction

Complex exponentials are some of the most important functions in our study of signals and systems. Their importance stems from their status as eigenfunctions of linear time invariant systems. Before proceeding, you should be familiar with complex numbers.

The Continuous Time Complex Exponential

Complex Exponentials

The complex exponential function will become a critical part of your study of signals and systems. Its general continuous form is written as

$$Ae^{st}$$

where $s = \sigma + i\omega$ is a complex number in terms of σ, the attenuation constant, and ω the angular frequency.

Euler's Formula

The mathematician Euler proved an important identity relating complex exponentials to trigonometric functions. Specifically, he discovered the eponymously named identity, Euler's formula, which states that

$$e^{jx} = \cos(x) + j\sin(x)$$

which can be proven as follows.

In order to prove Euler's formula, we start by evaluating the Taylor series for e^z about $z = 0$, which converges for all complex z, at $z = jx$. The result is

$$\begin{aligned} e^{jx} &= \sum_{k=0}^{\infty} \frac{(jx)^k}{k!} \\ &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} + j\sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \\ &= \cos(x) + j\sin(x) \end{aligned} \tag{1.8.1}$$

because the second expression contains the Taylor series for $\cos(x)$ and $\sin(x)$ about $t = 0$, which converge for all real x. Thus, the desired result is proven.

Choosing $x = \omega t$ this gives the result

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

which breaks a continuous time complex exponential into its real part and imaginary part. Using this formula, we can also derive the following relationships.

$$\cos(\omega t) = \frac{1}{2}e^{j\omega t} + \frac{1}{2}e^{-j\omega t}$$

$$\sin(\omega t) = \frac{1}{2j}e^{j\omega t} - \frac{1}{2j}e^{-j\omega t}$$

Continuous Time Phasors

It has been shown how the complex exponential with purely imaginary frequency can be broken up into its real part and its imaginary part. Now consider a general complex frequency $s = \sigma + \omega j$ where σ is the attenuation factor and ω is the frequency. Also consider a phase difference θ. It follows that

$$e^{(\sigma+j\omega)t+j\theta} = e^{\sigma t}(\cos(\omega t + \theta) + j\sin(\omega t + \theta))$$

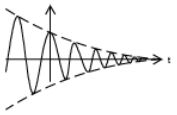
Thus, the real and imaginary parts of e^{st} appear below.

$$\mathrm{Re}\left\{e^{(\sigma+j\omega)t+j\theta}\right\} = e^{\sigma t}\cos(\omega t + \theta)$$

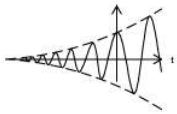
$$\text{Im}\left\{e^{(\sigma+j\omega)t+j\theta}\right\} = e^{\sigma t}\sin(\omega t+\theta)$$

Using the real or imaginary parts of complex exponential to represent sinusoids with a phase delay multiplied by real exponential is often useful and is called attenuated phasor notation.

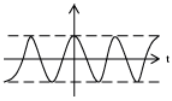
We can see that both the real part and the imaginary part have a sinusoid times a real exponential. We also know that sinusoids oscillate between one and negative one. From this it becomes apparent that the real and imaginary parts of the complex exponential will each oscillate within an envelope defined by the real exponential part.



(a)



(b)



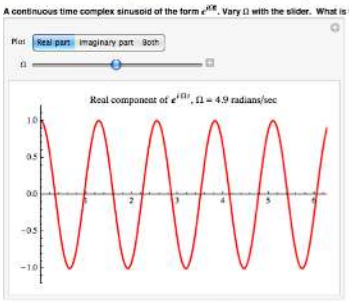
(c)

Figure 1.8.1: 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.

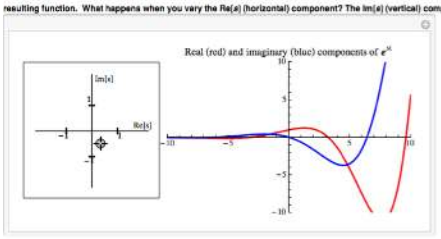
Complex Exponential Demonstration

Continuous-time complex sinusoids and complex exponentials

A continuous time complex sinusoid of the form $e^{j\Omega t}$. Vary Ω with the slider. What is the period of the function when $\Omega = 1$? 2? π? 2π?



A continuous time complex exponential of the form e^{st}. vary s by using the crosshair locator on the left part of the plot. The right side shows the resulting function. What happens when you vary the Re{s} (horizontal) component? The Im{s} (vertical) component?



https://eng.libretexts.org/@go/page/23114

Figure 1.8.2: Interact (when online) with a Mathematica CDF demonstrating the Continuous Time Complex Exponential. To Download, right-click and save target as .cdf.

Continuous Time Complex Exponential Summary

Continuous time complex exponentials are signals of great importance to the study of signals and systems. They can be related to sinusoids through Euler's formula, which identifies the real and imaginary parts of purely imaginary complex exponentials. Eulers formula reveals that, in general, the real and imaginary parts of complex exponentials are sinusoids multiplied by real exponentials. Thus, attenuated phasor notation is often useful in studying these signals.

This page titled 1.8: Continuous Time Complex Exponential is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

1.9: Discrete Time Complex Exponential

Introduction

Complex exponentials are some of the most important functions in our study of signals and systems. Their importance stems from their status as eigenfunctions of linear time invariant systems; as such, it can be both convenient and insightful to represent signals in terms of complex exponentials. Before proceeding, you should be familiar with complex numbers.

The Discrete Time Complex Exponential

Complex Exponentials

The complex exponential function will become a critical part of your study of signals and systems. Its general discrete form is written as

$$z^n$$

where z is a complex number. Recalling the polar expression of complex numbers, z can be expressed in terms of its magnitude $|z|$ and its angle (or argument) ω in the complex plane: $z = |z|e^{j\omega}$. Thus $z^n = (|z|)^n e^{j\omega n}$. In the context of complex exponentials, ω is referred to as frequency. For the time being, let's consider complex exponentials for which $|z| = 1$.

These discrete time complex exponentials have the following property, which will become evident through discussion of Euler's formula.

$$e^{j\omega n} = e^{j(\omega+2\pi)n}$$

Given this property, if we have a complex exponential with frequency $\omega + 2\pi$, then this signal "aliases" to a complex exponential with frequency ω, implying that the equation representations of discrete complex exponentials are not unique.

Euler's Formula

The mathematician Euler proved an important identity relating complex exponentials to trigonometric functions. Specifically, he discovered the eponymously named identity, Euler's formula, which states that for any real number x,

$$e^{jx} = \cos(x) + j\sin(x)$$

which can be proven as follows.

In order to prove Euler's formula, we start by evaluating the Taylor series for e^z about $z = 0$, which converges for all complex z, at $z = jx$. The result is

$$\begin{aligned} e^{jx} &= \sum_{k=0}^{\infty} \frac{(jx)^k}{k!} \\ &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} + j\sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \\ &= \cos(x) + j\sin(x) \end{aligned} \tag{1.9.1}$$

because the second expression contains the Taylor series for $\cos(x)$ and $\sin(x)$ about $t = 0$, which converge for all real x. Thus, the desired result is proven.

Choosing $x = \omega n$, we have:

$$e^{j\omega n} = \cos(\omega n) + j\sin(\omega n)$$

which breaks a discrete time complex exponential into its real part and imaginary part. Using this formula, we can also derive the following relationships.

$$\cos(\omega n) = \frac{1}{2}e^{j\omega n} + \frac{1}{2}e^{-j\omega n}$$

$$\sin(\omega n) = \frac{1}{2j}e^{j\omega n} - \frac{1}{2j}e^{-j\omega n}$$

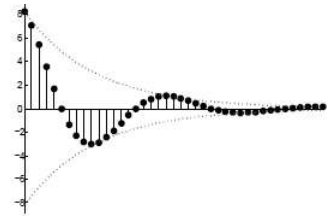
Real and Imaginary Parts of Complex Exponentials

Now let's return to the more general case of complex exponentials, z^n. Recall that $z^n = (|z|)^n e^{j\omega n}$. We can express this in terms of its real and imaginary parts:

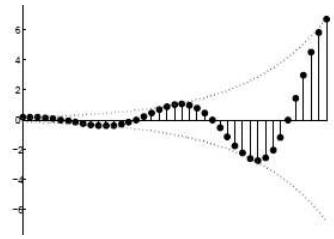
$$\mathrm{Re}\{z^n\} = (|z|)^n \cos(\omega n)$$

$$\mathrm{Im}\{z^n\} = (|z|)^n \sin(\omega n)$$

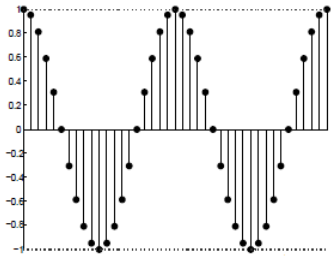
We see now that the magnitude of z establishes an exponential envelope to the signal, with ω controlling rate of the sinusoidal oscillation within the envelope.



(a)



(b)



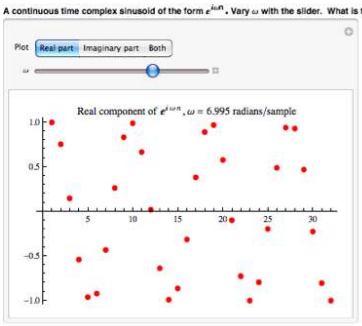
(c)

Figure 1.9.1: (a) If $|z| < 1$, we have the case of a decaying exponential envelope. (b) If $|z| > 1$, we have the case of a growing exponential envelope. (c) If $|z| = 1$, we have the case of a constant envelope.

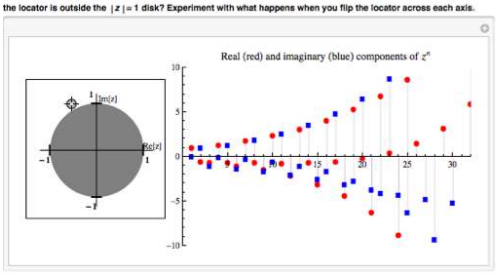
Discrete Complex Exponential Demonstration

Discrete-time complex sinusoids and complex exponentials

A continuous time complex sinusoid of the form $e^{i\omega n}$. Vary ω with the slider. What is the period of the function when $\omega = 1$? 2? π? 2π?



A continuous time complex exponential of the form z^n. Vary z by using the crosshair locator on the left part of the plot. The right side shows the resulting function. What happens when you vary the Re[z] (horizontal) component? The Im[z] (vertical) component? What happens when the locator is outside the $|z| = 1$ disk? Experiment with what happens when you flip the locator across each axis.



https://eng.libretexts.org/@go/page/23115

Figure 1.9.2: Interact (when online) with a Mathematica CDF demonstrating the Discrete Time Complex Exponential. To Download, right-click and save target as .cdf.

Discrete Time Complex Exponential Summary

Discrete time complex exponentials are signals of great importance to the study of signals and systems. They can be related to sinusoids through Euler's formula, which identifies the real and imaginary parts of complex exponentials. Eulers formula reveals that, in general, the real and imaginary parts of complex exponentials are sinusoids multiplied by real exponentials.

This page titled 1.9: Discrete Time Complex Exponential is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

2: Introduction to Systems

2.1: System Classifications and Properties
2.2: Linear Time Invariant Systems

This page titled 2: Introduction to Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

2.1: System Classifications and Properties

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 distinguish one system from another. Understanding these basic differences between systems, and their properties, will be a fundamental concept used in all signal and system courses. Once a set of systems can be identified as sharing particular properties, one no longer has to reprove a certain characteristic of a system each time, but it can simply be known due to the the system classification.

Classification of Systems

Continuous vs. Discrete

One of the most important distinctions to understand is the difference between discrete time and continuous time systems. A system in which the input signal and output signal both have continuous domains is said to be a continuous system. One in which the input signal and output signal both have discrete domains is said to be a discrete system. Of course, it is possible to conceive of signals that belong to neither category, such as systems in which sampling of a continuous time signal or reconstruction from a discrete time signal take place.

Linear vs. Nonlinear

A linear system is any system that obeys the properties of scaling (first order homogeneity) and superposition (additivity) further described below. A nonlinear system is any system that does not have at least one of these properties.

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

$$H(kf(t)) = kH(f(t))$$

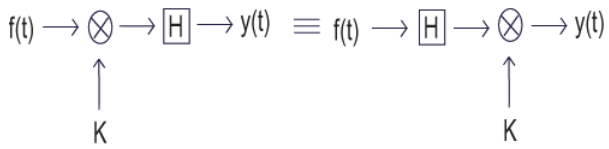


Figure 2.1.1: A block diagram demonstrating the scaling property of linearity

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

$$H\left(f_1(t) + f_2(t)\right) = H\left(f_1(t)\right) + H\left(f_2(t)\right)$$

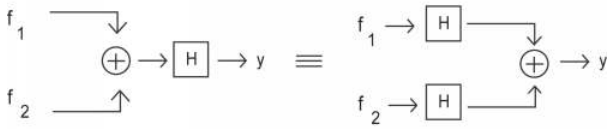


Figure 2.1.2: A block diagram demonstrating the superposition property of linearity

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

$$H\left(k_1 f_1(t) + k_2 f_2(t)\right) = k_1 H\left(f_1(t)\right) + k_2 H\left(f_2(t)\right)$$

Time Invariant vs. Time Varying

A system is said to be time invariant if it commutes with the parameter shift operator defined by $S_T(f(t)) = f(t - T)$ for all T, which is to say

$$HS_T = S_T H$$

for all real T. Intuitively, that means that for any input function that produces some output function, any time shift of that input function will produce an output function identical in every way except that it is shifted by the same amount. Any system that does not have this property is said to be time varying.

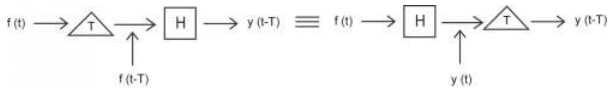


Figure 2.1.3: This block diagram shows what the condition for time invariance. The output is the same whether the delay is put on the input or the output.

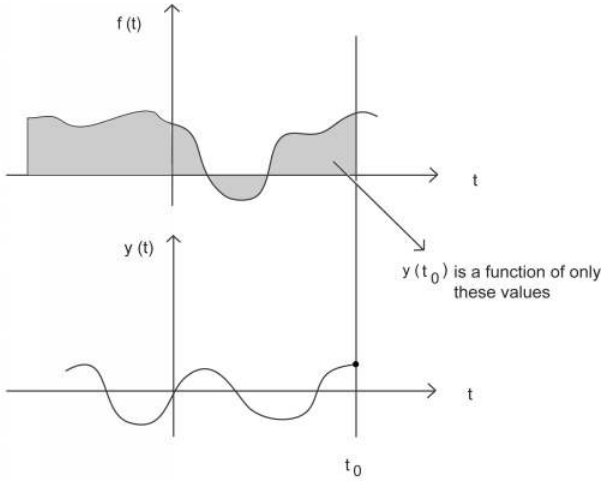
Causal vs. Noncausal

A causal system is one in which the output depends only on current or past inputs, but not future inputs. Similarly, an anticausal system is one in which the output depends only on current or future inputs, but not past inputs. Finally, a noncausal system is one in which the output depends on both past and 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 pixel positions to the left and right (the "future") of the current position on the image, and we would not necessarily have a causal system.



(a)



(b)

Figure 2.1.4: (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.

Stable vs. Unstable

There are several definitions of stability, but the one that will be used most frequently in this course will be bounded input, bounded output (BIBO) stability. In this context, a stable system is one in which the output is bounded if the input is also bounded. Similarly, an unstable system is one in which at least one bounded input produces an unbounded output.

Representing this mathematically, 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

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

whenever we have an input to the system that satisfies

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

M_x and M_y both represent a set of finite positive numbers and these relationships hold for all of t. Otherwise, the system is unstable.

System Classifications Summary

This module describes just some of the many ways in which systems can be classified. Systems can be continuous time, discrete time, or neither. They can be linear or nonlinear, time invariant or time varying, and stable or unstable. We can also divide them based on their causality properties. There are other ways to classify systems, such as use of memory, that are not discussed here but will be described in subsequent modules.

This page titled 2.1: System Classifications and Properties is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

2.2: Linear Time Invariant Systems

Introduction

Linearity and time invariance are two system properties that greatly simplify the study of systems that exhibit them. In our study of signals and systems, we will be especially interested in systems that demonstrate both of these properties, which together allow the use of some of the most powerful tools of signal processing.

Linear Time Invariant Systems

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.

Linear Scaling

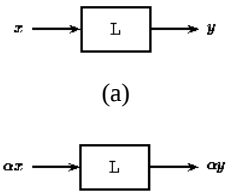


Figure 2.2.1

In Figure 2.2.1(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 2.2.1(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.

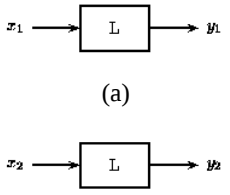


Figure 2.2.2

Superposition Principle

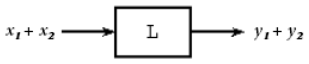
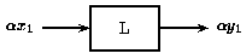
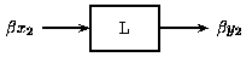


Figure 2.2.3: If Figure 2.2.2 is true, then the principle of superposition says that Figure 2.2.3 is true as well. This holds for linear systems.

That is, if Figure 2.2.2 is true, then Figure 2.2.3 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:



(a)



(b)

Figure 2.2.4

Superposition Principle with Linear Scaling

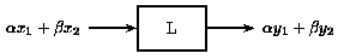


Figure 2.2.5: Given Figure 2.2.4 for a linear system, Figure 2.2.5 holds as well.

Example 2.2.1

Consider the system H_1 in which

$$H_1(f(t)) = tf(t)$$

for all signals f, g and scalars a, b

$$H_1(af(t) + bg(t))) = t(af(t) + bg(t)) = atf(t) + btg(t) = aH_1(f(t)) + bH_1(g(t))$$

for all real t. Thus, H_1 is a linear system.

Example 2.2.2

Consider the system H_2 in which

$$H_2(f(t)) = (f(t))^2$$

for all signals f. Because

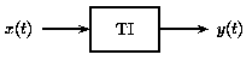
$$H_2(2t) = 4t^2 \neq 2t^2 = 2H_2(t)$$

for nonzero t, H_2 is not a linear system.

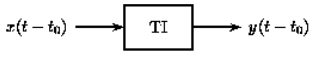
Time Invariant Systems

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

Time-Invariant Systems



(a)



(b)

Figure 2.2.6: Figure 2.2.6(a) shows an input at time t while Figure 2.2.6(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 2.2.6(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.

Example 2.2.3

Consider the system H_1 in which

$$H_1(f(t)) = tf(t)$$

for all signals f. Because

$$S_T\left(H_1(f(t))\right) = S_T(tf(t)) = (t-T)f(t-T) \neq tf(t-T) = H_1(f(t-T)) = H_1\left(S_T(f(t))\right)$$

for nonzero T, H_1 is not a time invariant system.

Example 2.2.4

Consider the system H_2 in which

$$H_2(f(t)) = (f(t))^2$$

for all signals f. For all real T and signals f,

$$S_T\left(H_2(f(t))\right) = S_T\left(f(t)^2\right) = (f(t-T))^2 = H_2(f(t-T)) = H_2\left(S_T(f(t))\right)$$

for all real t. Thus, H_2 is a time invariant system.

Linear Time Invariant Systems

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

Linear Time-Invariant Systems



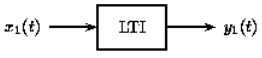
(a)



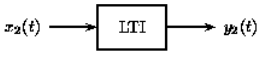
(b)

Figure 2.2.7: This is a combination of the two cases above. Since the input to Figure 2.2.7(b) is a scaled, time-shifted version of the input in Figure 2.2.7(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.



(a)



(b)

Figure 2.2.8

Superposition in Linear Time-Invariant Systems

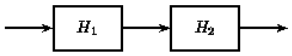


Figure 2.2.9: The principle of superposition applied to LTI systems

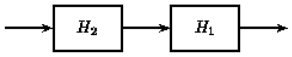
LTI Systems in Series

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)



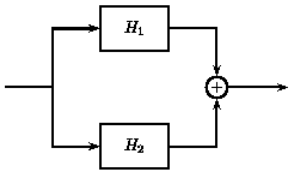
(b)

Figure 2.2.10: The order of cascaded LTI systems can be interchanged without changing the overall effect.

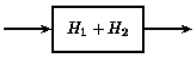
LTI Systems in Parallel

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 2.2.11: Parallel systems can be condensed into the sum of systems.

Example 2.2.5

Consider the system H_3 in which

$$H_3(f(t)) = 2f(t)$$

for all signals f. Given any two signals f, g and scalars a, b

$$H_3(af(t)+bg(t))) = 2(af(t)+bg(t)) = a2f(t)+b2g(t) = aH_3(f(t))+bH_3(g(t))$$

for all real t. Thus, H_3 is a linear system. For all real T and signals f,

$$S_T\left(H_3(f(t))\right) = S_T(2f(t)) = 2f(t-T) = H_3(f(t-T)) = H_3\left(S_T(f(t))\right)$$

for all real t. Thus, H_3 is a time invariant system. Therefore, H_3 is a linear time invariant system.

Example 2.2.6

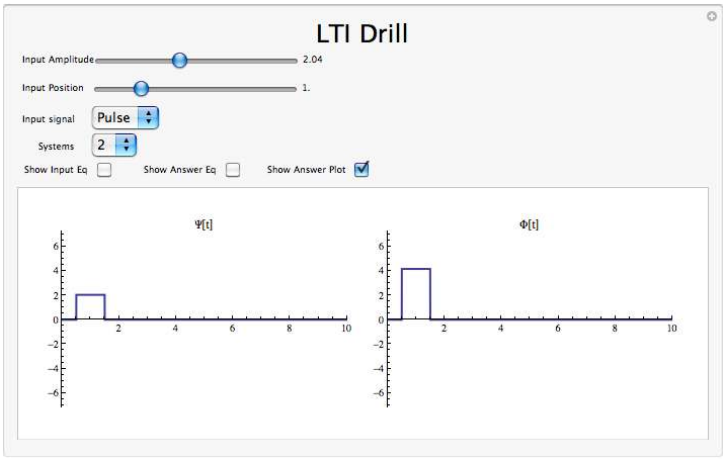
As has been previously shown, each of the following systems are not linear or not time invariant.

$$H_1(f(t)) = tf(t)$$

$$H_2(f(t)) = (f(t))^2$$

Thus, they are not linear time invariant systems.

Linear Time Invariant Demonstration



Interact(when online) with the Mathematica CDF above demonstrating Linear Time Invariant systems. To download, right click and save file as .cdf.

LTI Systems Summary

Two very important and useful properties of systems have just been described in detail. The first of these, linearity, allows us the knowledge that a sum of input signals produces an output signal that is the summed original output signals and that a scaled input signal produces an output signal scaled from the original output signal. The second of these, time invariance, ensures that time shifts commute with application of the system. In other words, the output signal for a time shifted input is the same as the output

signal for the original input signal, except for an identical shift in time. Systems that demonstrate both linearity and time invariance, which are given the acronym LTI systems, are particularly simple to study as these properties allow us to leverage some of the most powerful tools in signal processing.

This page titled 2.2: Linear Time Invariant Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

3: Time Domain Analysis of Continuous Time Systems

3.1: Continuous Time Systems
3.2: Continuous Time Impulse Response
3.3: Continuous Time Convolution
3.4: Properties of Continuous Time Convolution
3.5: Eigenfunctions of Continuous Time LTI Systems
3.6: BIBO Stability of Continuous Time Systems
3.7: Linear Constant Coefficient Differential Equations
3.8: Solving Linear Constant Coefficient Differential Equations

---

This page titled 3: Time Domain Analysis of Continuous Time Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

3.1: Continuous Time Systems

Introduction

As you already now know, a continuous time system operates on a continuous time signal input and produces a continuous time signal output. There are numerous examples of useful continuous time systems in signal processing as they essentially describe the world around us. The class of continuous time systems that are both linear and time invariant, known as continuous time LTI systems, is of particular interest as the properties of linearity and time invariance together allow the use of some of the most important and powerful tools in signal processing.

Continuous Time Systems

Linearity and Time Invariance

A system H is said to be linear if it satisfies two important conditions. The first, additivity, states for every pair of signals x, y that $H(x+y) = H(x) + H(y)$. The second, homogeneity of degree one, states for every signal x and scalar a we have $H(ax) = aH(x)$. It is clear that these conditions can be combined together into a single condition for linearity. Thus, a system is said to be linear if for every signals x, y and scalars a, b we have that

$$H(ax + by) = aH(x) + bH(y)$$

Linearity is a particularly important property of systems as it allows us to leverage the powerful tools of linear algebra, such as bases, eigenvectors, and eigenvalues, in their study.

A system H is said to be time invariant if a time shift of an input produces the corresponding shifted output. In other, more precise words, the system H commutes with the time shift operator S_T for every $T \in \mathbb{R}$. That is,

$$S_T H = H S_T$$

Time invariance is desirable because it eases computation while mirroring our intuition that, all else equal, physical systems should react the same to identical inputs at different times.

When a system exhibits both of these important properties it allows for a more straigtforward analysis than would otherwise be possible. As will be explained and proven in subsequent modules, computation of the system output for a given input becomes a simple matter of convolving the input with the system's impulse response signal. Also proven later, the fact that complex exponential are eigenvectors of linear time invariant systems will enable the use of frequency domain tools such as the various Fourier transforms and associated transfer functions, to describe the behavior of linear time invariant systems.

Example 3.1.1

Consider the system H in which

$$H(f(t)) = 2f(t)$$

for all signals f. Given any two signals f, g and scalars a, b

$$H(af(t) + bg(t))) = 2(af(t) + bg(t)) = a2f(t) + b2g(t) = aH(f(t)) + bH(g(t))$$

for all real t. Thus, H is a linear system. For all real T and signals f,

$$S_T(H(f(t))) = S_T(2f(t)) = 2f(t-T) = H(f(t-T)) = H\left(S_T(f(t))\right)$$

for all real t. Thus, H is a time invariant system. Therefore, H is a linear time invariant system.

Differential Equation Representation

It is often useful to to describe systems using equations involving the rate of change in some quantity. For continuous time systems, such equations are called differential equations. One important class of differential equations is the set of linear constant coefficient ordinary differential equations, which are described in more detail in subsequent modules.

Example 3.1.2

Consider the series RLC circuit shown in Figure 3.1.1. This system can be modeled using differential equations. We can use the voltage equations for each circuit element and Kirchoff's voltage law to write a second order linear constant coefficient differential equation describing the charge on the capacitor.

The voltage across the battery is simply V. The voltage across the capacitor is $\frac{1}{C}q$. The voltage across the resistor is $R\frac{dq}{dt}$. Finally, the voltage across the inductor is $L\frac{d^2q}{dt^2}$. Therefore, by Kirchoff's voltage law, it follows that

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q = V$$

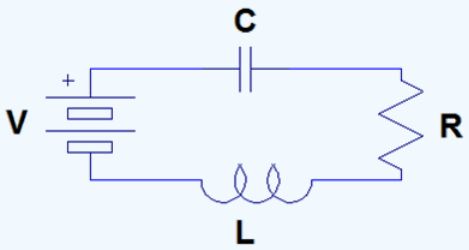


Figure 3.1.1: A series RLC circuit.

Continuous Time Systems Summary

Many useful continuous time systems will be encountered in a study of signals and systems. This course is most interested in those that demonstrate both the linearity property and the time invariance property, which together enable the use of some of the most powerful tools of signal processing. It is often useful to describe them in terms of rates of change through linear constant coefficient ordinary differential equations.

This page titled 3.1: Continuous Time Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

3.2: Continuous Time Impulse Response

Introduction

The output of an LTI system is completely determined by the input and the system's response to a unit impulse.

System Output

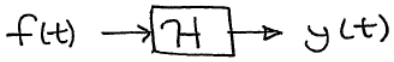


Figure 3.2.1: We can determine the system's output, $y(t)$, if we know the system's impulse response, $h(t)$, and the input, $f(t)$.

The output for a unit impulse input is called the impulse response.

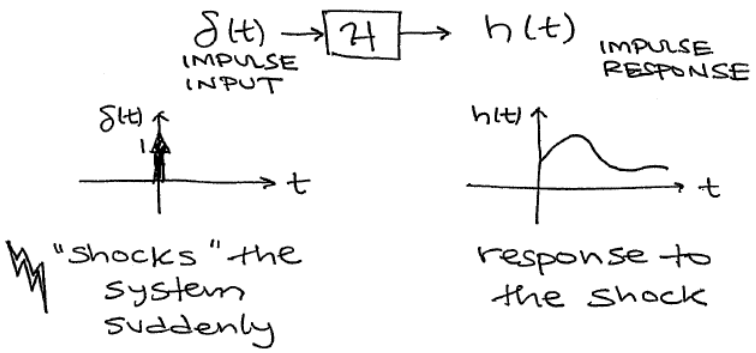


Figure 3.2.2

Example Approximate Impulses

1. Hammer blow to a structure
2. Hand clap or gun blast in a room
3. Air gun blast underwater

LTI Systems and Impulse Responses

Finding System Outputs

By the sifting property of impulses, any signal can be decomposed in terms of an integral of shifted, scaled impulses.

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

$\delta(t-\tau)$ peaks up where $t = \tau$.

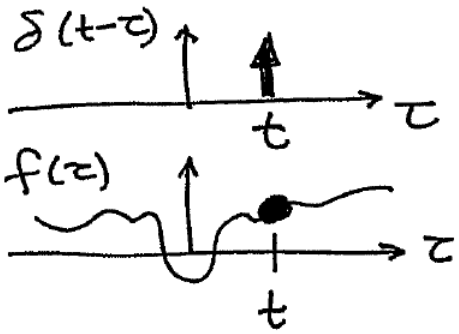


Figure 3.2.3

Since we know the response of the system to an impulse and any signal can be decomposed into impulses, all we need to do to find the response of the system to any signal is to decompose the signal into impulses, calculate the system's output for every impulse

and add the outputs back together. This is the process known as **Convolution**. Since we are in **Continuous Time**, this is the **Continuous Time Convolution Integral**.

Finding Impulse Responses

Theory:

1. Solve the system's **differential equation** for $y(t)$ with $f(t) = \delta(t)$
2. Use the **Laplace transform**

Practice:

1. Apply an **impulse-like input signal** to the system and measure the output
2. Use **Fourier methods**

We will assume that $h(t)$ is given for now.

- The goal now is to compute the output $y(t)$ given the impulse response $h(t)$ and the input $f(t)$.

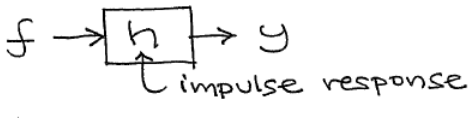


Figure 3.2.4

Impulse Response Summary

When a system is "shocked" by a delta function, it produces an output known as its impulse response. For an LTI system, the impulse response completely determines the output of the system given any arbitrary input. The output can be found using continuous time convolution.

This page titled 3.2: Continuous Time Impulse Response is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

3.3: Continuous Time Convolution

Introduction

Convolution, one of the most important concepts in electrical engineering, can be used to determine the output a system produces for a given input signal. It can be shown that a linear time invariant system is completely characterized by its impulse response. The sifting property of the continuous time impulse function tells us that the input signal to a system can be represented as an integral of scaled and shifted impulses and, therefore, as the limit of a sum of scaled and shifted approximate unit impulses. Thus, by linearity, it would seem reasonable to compute of the output signal as the limit of a sum of scaled and shifted unit impulse responses and, therefore, as the integral of a scaled and shifted impulse response. That is exactly what the operation of convolution accomplishes. Hence, convolution can be used to determine a linear time invariant system's output from knowledge of the input and the impulse response.

Convolution and Circular Convolution

Convolution

Operation Definition

Continuous time convolution is an operation on two continuous time signals defined by the integral

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

for all signals f, g defined on $\mathbb{R}$. It is important to note that the operation of convolution is commutative, meaning that

$$f^*g = g^*f$$

for all signals f, g defined on $\mathbb{R}$. Thus, the convolution operation could have been just as easily stated using the equivalent definition

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

for all signals f, g defined on $\mathbb{R}$. Convolution has several other important properties not listed here but explained and derived in a later module.

Definition Motivation

The above operation definition has been chosen to be particularly useful in the study of linear time invariant systems. In order to see this, consider a linear time invariant system H with unit impulse response h. Given a system input signal x we would like to compute the system output signal $H(x)$. First, we note that the input can be expressed as the convolution

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

by the sifting property of the unit impulse function. Writing this integral as the limit of a summation,

$$x(t) = \lim_{\Delta \to 0} \sum_n x(n\Delta)\delta_\Delta(t-n\Delta)\Delta$$

where

$$\delta_\Delta(t) = \begin{cases} 1/\Delta & 0 \le t < \Delta \\ 0 & \text{otherwise} \end{cases}$$

approximates the properties of $\delta(t)$. By linearity

$$Hx(t) = \lim_{\Delta \to 0} \sum_n x(n\Delta)H\delta_\Delta(t-n\Delta)\Delta$$

which evaluated as an integral gives

$$Hx(t) = \int_{-\infty}^{\infty} x(\tau) H\delta(t-\tau) d\tau$$

Since $H\delta(t-\tau)$ is the shifted unit impulse response $h(t-\tau)$, this gives the result

$$Hx(t) = \int_{-\infty}^{\infty} x(\tau) h(t-\tau) d\tau = (x^* h)(t)$$

Hence, convolution has been defined such that the output of a linear time invariant system is given by the convolution of the system input with the system unit impulse response.

Graphical Intuition

It is often helpful to be able to visualize the computation of a convolution in terms of graphical processes. Consider the convolution of two functions f, g given by

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

The first step in graphically understanding the operation of convolution is to plot each of the functions. Next, one of the functions must be selected, and its plot reflected across the $\tau = 0$ axis. For each real t, that same function must be shifted left by t. The product of the two resulting plots is then constructed. Finally, the area under the resulting curve is computed.

Example 3.3.1

Recall that the impulse response for the capacitor voltage in a series RC circuit is given by

$$h(t) = \frac{1}{RC} e^{-t/RC} u(t),$$

and consider the response to the input voltage

$$x(t) = u(t).$$

We know that the output for this input voltage is given by the convolution of the impulse response with the input signal

$$y(t) = x(t) * h(t)$$

We would like to compute this operation by beginning in a way that minimizes the algebraic complexity of the expression. Thus, since $x(t) = u(t)$ is the simpler of the two signals, it is desirable to select it for time reversal and shifting. Thus, we would like to compute

$$y(t) = \int_{-\infty}^{\infty} \frac{1}{RC} e^{-\tau/RC} u(\tau) u(t-\tau) d\tau$$

The step functions can be used to further simplify this integral by narrowing the region of integration to the nonzero region of the integrand. Therefore,

$$y(t) = \int_{0}^{\max\{0,t\}} \frac{1}{RC} e^{-\tau/RC} d\tau$$

Hence, the output is

$$y(t) = \begin{cases} 0 & t \leq 0 \\ 1 - e^{-t/RC} & t > 0 \end{cases}$$

which can also be written as

$$y(t) = \left(1 - e^{-t/RC}\right) u(t).$$

Circular Convolution

Continuous time circular convolution is an operation on two finite length or periodic continuous time signals defined by the integral

$$(f * g)(t) = \int_0^T \hat{f}(\tau)\hat{g}(t-\tau)d\tau$$

for all signals f, g defined on $\mathbb{R}[0,T]$ where $\hat{f}$, $\hat{g}$ are periodic extensions of f and g. It is important to note that the operation of circular convolution is commutative, meaning that

$$f^* g = g^* f$$

for all signals f, g defined on $\mathbb{R}[0,T]$. Thus, the circular convolution operation could have been just as easily stated using the equivalent definition

$$(f * g)(t) = \int_0^T \hat{f}(t-\tau)\hat{g}(\tau)d\tau$$

for all signals f, g defined on $\mathbb{R}[0,T]$ where $\hat{f}$, $\hat{g}$ are periodic extensions of f and g. Circular convolution has several other important properties not listed here but explained and derived in a later module.

Alternatively, continuous time circular convolution can be expressed as the sum of two integrals given by

$$(f * g)(t) = \int_0^t f(\tau)g(t-\tau)d\tau + \int_t^T f(\tau)g(t-\tau+T)d\tau$$

for all signals f, g defined on $\mathbb{R}[0,T]$.

Meaningful examples of computing continuous time circular convolutions in the time domain would involve complicated algebraic manipulations dealing with the wrap around behavior, which would ultimately be more confusing than helpful. Thus, none will be provided in this section. However, continuous time circular convolutions are more easily computed using frequency domain tools as will be shown in the continuous time Fourier series section.

Definition Motivation

The above operation definition has been chosen to be particularly useful in the study of linear time invariant systems. In order to see this, consider a linear time invariant system H with unit impulse response h. Given a finite or periodic system input signal x we would like to compute the system output signal $H(x)$. First, we note that the input can be expressed as the circular convolution

$$x(t) = \int_0^T \hat{x}(\tau)\hat{\delta}(t-\tau)d\tau$$

by the sifting property of the unit impulse function. Writing this integral as the limit of a summation,

$$x(t) = \lim_{\Delta \to 0} \sum_n \hat{x}(n\Delta)\hat{\delta}_\Delta(t-n\Delta)\Delta$$

where

$$\delta_\Delta(t) = \begin{cases} 1/\Delta & 0 \le t < \Delta \\ 0 & \text{otherwise} \end{cases}$$

approximates the properties of $\delta(t)$. By linearity

$$Hx(t) = \lim_{\Delta \to 0} \sum_n \hat{x}(n\Delta)H\hat{\delta}_\Delta(t-n\Delta)\Delta$$

which evaluated as an integral gives

$$Hx(t) = \int_0^T \hat{x}(\tau)H\hat{\delta}(t-\tau)d\tau$$

Since $H\delta(t-\tau)$ is the shifted unit impulse response $h(t-\tau)$, this gives the result

$$Hx(t) = \int_0^T \hat{x}(\tau)\hat{h}(t-\tau)d\tau = (x * h)(t).$$

Hence, circular convolution has been defined such that the output of a linear time invariant system is given by the convolution of the system input with the system unit impulse response.

Graphical Intuition

It is often helpful to be able to visualize the computation of a circular convolution in terms of graphical processes. Consider the circular convolution of two finite length functions f, g given by

$$(f^* g)(t) = \int_0^T \hat{f}(\tau)\hat{g}(t-\tau)d\tau = \int_0^T \hat{f}(t-\tau)\hat{g}(\tau)d\tau$$

The first step in graphically understanding the operation of convolution is to plot each of the periodic extensions of the functions. Next, one of the functions must be selected, and its plot reflected across the $\tau = 0$ axis. For each $t \in \mathbb{R}[0, T]$, that same function must be shifted left by t. The product of the two resulting plots is then constructed. Finally, the area under the resulting curve on $\mathbb{R}[0, T]$ is computed.

Convolution Demonstration

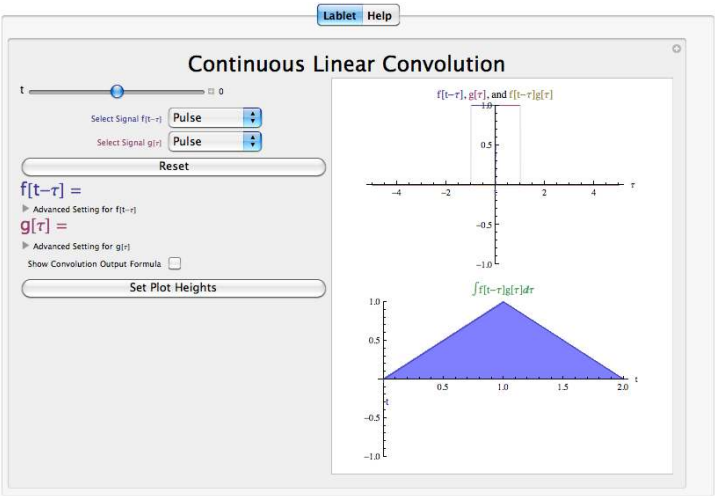


Figure 3.3.1: Interact (when online) with a Mathematica CDF demonstrating Convolution. To Download, right-click and save target as .cdf.

Convolution Summary

Convolution, one of the most important concepts in electrical engineering, can be used to determine the output signal of a linear time invariant system for a given input signal with knowledge of the system's unit impulse response. The operation of continuous time convolution is defined such that it performs this function for infinite length continuous time signals and systems. The

operation of continuous time circular convolution is defined such that it performs this function for finite length and periodic continuous time signals. In each case, the output of the system is the convolution or circular convolution of the input signal with the unit impulse response.

This page titled 3.3: Continuous Time Convolution is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

3.4: Properties of Continuous Time Convolution

Introduction

We have already shown the important role that continuous time convolution plays in signal processing. This section provides discussion and proof of some of the important properties of continuous time convolution. Analogous properties can be shown for continuous time circular convolution with trivial modification of the proofs provided except where explicitly noted otherwise.

Continuous Time Convolution Properties

Associativity

The operation of convolution is associative. That is, for all continuous time signals x_1, x_2, x_3 the following relationship holds.

$$x_1 * (x_2 * x_3) = (x_1 * x_2) * x_3$$

In order to show this, note that

$$\begin{aligned}
(x_1 * (x_2 * x_3))(t) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1(\tau_1)\, x_2(\tau_2)\, x_3((t-\tau_1)-\tau_2)\, d\tau_2 d\tau_1 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1(\tau_1)\, x_2((\tau_1+\tau_2)-\tau_1)\, x_3(t-(\tau_1+\tau_2))\, d\tau_2 d\tau_1 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1(\tau_1)\, x_2(\tau_3-\tau_1)\, x_3(t-\tau_3)\, d\tau_1 d\tau_3 \\
&= ((x_1 * x_2) * x_3)(t)
\end{aligned} \tag{3.4.1}$$

proving the relationship as desired through the substitution $\tau_3 = \tau_1 + \tau_2$.

Commutativity

The operation of convolution is commutative. That is, for all continuous time signals x_1, x_2 the following relationship holds.

$$x_1 * x_2 = x_2 * x_1$$

In order to show this, note that

$$\begin{aligned}
(x_1 * x_2)(t) &= \int_{-\infty}^{\infty} x_1(\tau_1)\, x_2(t-\tau_1)\, d\tau_1 \\
&= \int_{-\infty}^{\infty} x_1(t-\tau_2)\, x_2(\tau_2)\, d\tau_2 \\
&= (x_2 * x_1)(t)
\end{aligned}$$

proving the relationship as desired through the substitution $\tau_2 = t - \tau_1$.

Distributivity

The operation of convolution is distributive over the operation of addition. That is, for all continuous time signals x_1, x_2, x_3 the following relationship holds.

$$x_1 * (x_2 + x_3) = x_1 * x_2 + x_1 * x_3$$

In order to show this, note that

$$\begin{aligned}
(x_1 * (x_2 + x_3))(t) &= \int_{-\infty}^{\infty} x_1(\tau)\,(x_2(t-\tau) + x_3(t-\tau))\, d\tau \\
&= \int_{-\infty}^{\infty} x_1(\tau) x_2(t-\tau) d\tau + \int_{-\infty}^{\infty} x_1(\tau) x_3(t-\tau) d\tau \\
&= (x_1 * x_2 + x_1 * x_3)(t)
\end{aligned} \tag{3.4.2}$$

proving the relationship as desired.

Multilinearity

The operation of convolution is linear in each of the two function variables. Additivity in each variable results from distributivity of convolution over addition. Homogenity of order one in each variable results from the fact that for all continuous time signals x_1, x_2 and scalars a the following relationship holds.

$$a\left(x_1 * x_2\right) = \left(a x_1\right) * x_2 = x_1 * \left(a x_2\right)$$

In order to show this, note that

$$\begin{aligned}
\left(a\left(x_1 * x_2\right)\right)(t) &= a \int_{-\infty}^{\infty} x_1(\tau) x_2(t-\tau) d\tau \\
&= \int_{-\infty}^{\infty} \left(a x_1(\tau)\right) x_2(t-\tau) d\tau \\
&= \left(\left(a x_1\right) * x_2\right)(t) \\
&= \int_{-\infty}^{\infty} x_1(\tau)\left(a x_2(t-\tau)\right) d\tau \\
&= \left(x_1 * \left(a x_2\right)\right)(t)
\end{aligned} \tag{3.4.3}$$

proving the relationship as desired.

Conjugation

The operation of convolution has the following property for all continuous time signals x_1, x_2.

$$\overline{x_1^* x_2} = \overline{x_1} * \overline{x_2}$$

In order to show this, note that

$$\begin{aligned}
(\overline{x_1 * x_2})(t) &= \overline{\int_{-\infty}^{\infty} x_1(\tau) x_2(t-\tau) d\tau} \\
&= \int_{-\infty}^{\infty} \overline{x_1(\tau) x_2(t-\tau)} d\tau \\
&= \int_{-\infty}^{\infty} \overline{x_1}(\tau) \overline{x_2}(t-\tau) d\tau \\
&= (\overline{x_1} * \overline{x_2})(t)
\end{aligned} \tag{3.4.4}$$

proving the relationship as desired.

Time Shift

The operation of convolution has the following property for all continuous time signals x_1, x_2 where S_T is the time shift operator.

$$S_T\left(x_1 * x_2\right) = \left(S_T x_1\right) * x_2 = x_1 * \left(S_T x_2\right)$$

In order to show this, note that

$$\begin{aligned}
S_T\left(x_1 * x_2\right)(t) &= \int_{-\infty}^{\infty} x_2(\tau) x_1((t-T)-\tau) d\tau \\
&= \int_{-\infty}^{\infty} x_2(\tau) S_T x_1(t-\tau) d\tau \\
&= \left(\left(S_T x_1\right) * x_2\right)(t) \\
&= \int_{-\infty}^{\infty} x_1(\tau) x_2((t-T)-\tau) d\tau \\
&= \int_{-\infty}^{\infty} x_1(\tau) S_T x_2(t-\tau) d\tau \\
&= x_1 * \left(S_T x_2\right)(t)
\end{aligned} \tag{3.4.5}$$

proving the relationship as desired.

Differentiation

The operation of convolution has the following property for all continuous time signals x_1, x_2.

$$\frac{d}{dt}\left(x_1 * x_2\right)(t) = \left(\frac{dx_1}{dt} * x_2\right)(t) = \left(x_1 * \frac{dx_2}{dt}\right)(t)$$

In order to show this, note that

$$\begin{aligned}\frac{d}{dt}\left(x_1 * x_2\right)(t) &= \int_{-\infty}^{\infty} x_2(\tau)\frac{d}{dt}x_1(t-\tau)d\tau \\ &= \left(\frac{dx_1}{dt} * x_2\right)(t) \\ &= \int_{-\infty}^{\infty} x_1(\tau)\frac{d}{dt}x_2(t-\tau)d\tau \\ &= \left(x_1 * \frac{dx_2}{dt}\right)(t)\end{aligned} \tag{3.4.6}$$

proving the relationship as desired.

Impulse Convolution

The operation of convolution has the following property for all continuous time signals x where $delta$ is the Dirac delta funciton.

$$x * \delta = x$$

In order to show this, note that

$$\begin{aligned}(x * \delta)(t) &= \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau \\ &= x(t)\int_{-\infty}^{\infty} \delta(t-\tau)d\tau \\ &= x(t)\end{aligned} \tag{3.4.7}$$

proving the relationship as desired.

Width

The operation of convolution has the following property for all continuous time signals x_1, x_2 where Duration(x) gives the duration of a signal x.

$$\text{Duration}\left(x_1 * x_2\right) = \text{ Duration}\left(x_1\right) + \text{ Duration}\left(x_2\right)$$

In order to show this informally, note that $(x_1 * x_2)(t)$ is nonzero for all tt for which there is a τ such that $x_1(\tau)x_2(t-\tau)$ is nonzero. When viewing one function as reversed and sliding past the other, it is easy to see that such a τ exists for all t on an interval of length Duration(x_1) + Duration(x_2). Note that this is not always true of circular convolution of finite length and periodic signals as there is then a maximum possible duration within a period.

Convolution Properties Summary

As can be seen the operation of continuous time convolution has several important properties that have been listed and proven in this module. With slight modifications to proofs, most of these also extend to continuous time circular convolution as well and the cases in which exceptions occur have been noted above. These identities will be useful to keep in mind as the reader continues to study signals and systems.

This page titled 3.4: Properties of Continuous Time Convolution is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

3.5: Eigenfunctions of Continuous Time LTI Systems

Introduction

Prior to reading this module, the reader should already have some experience with linear algebra and should specifically be familiar with the eigenvectors and eigenvalues of linear operators. A linear time invariant system is a linear operator defined on a function space that commutes with every time shift operator on that function space. Thus, we can also consider the eigenvector functions, or eigenfunctions, of a system. It is particularly easy to calculate the output of a system when an eigenfunction is the input as the output is simply the eigenfunction scaled by the associated eigenvalue. As will be shown, continuous time complex exponentials serve as eigenfunctions of linear time invariant systems operating on continuous time signals.

Eigenfunctions of LTI Systems

Consider a linear time invariant system H with impulse response h operating on some space of infinite length continuous time signals. Recall that the output $H(x(t))$ of the system for a given input $x(t)$ is given by the continuous time convolution of the impulse response with the input

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

Now consider the input $x(t) = e^{st}$ where $s \in \mathbb{C}$. Computing the output for this input,

$$\begin{aligned} H\left(e^{st}\right) &= \int_{-\infty}^{\infty} h(\tau)e^{s(t-\tau)}d\tau \\ &= \int_{-\infty}^{\infty} h(\tau)e^{st}e^{-s\tau}d\tau \\ &= e^{st}\int_{-\infty}^{\infty} h(\tau)e^{-s\tau}d\tau \end{aligned} \tag{3.5.1}$$

Thus,

$$H\left(e^{st}\right) = \lambda_s e^{st}$$

where

$$\lambda_s = \int_{-\infty}^{\infty} h(\tau)e^{-s\tau}d\tau$$

is the eigenvalue corresponding to the eigenvector e^{st}.

There are some additional points that should be mentioned. Note that, there still may be additional eigenvalues of a linear time invariant system not described by e^{st} for some $s \in \mathbb{C}$. Furthermore, the above discussion has been somewhat formally loose as e^{st} may or may not belong to the space on which the system operates. However, for our purposes, complex exponentials will be accepted as eigenvectors of linear time invariant systems. A similar argument using continuous time circular convolution would also hold for spaces finite length signals.

Eigenfunction of LTI Systems Summary

As has been shown, continuous time complex exponential are eigenfunctions of linear time invariant systems operating on continuous time signals. Thus, it is particularly simple to calculate the output of a linear time invariant system for a complex exponential input as the result is a complex exponential output scaled by the associated eigenvalue. Consequently, representations of continuous time signals in terms of continuous time complex exponentials provide an advantage when studying signals. As will be explained later, this is what is accomplished by the continuous time Fourier transform and continuous time Fourier series, which apply to aperiodic and periodic signals respectively.

This page titled 3.5: Eigenfunctions of Continuous Time LTI Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

3.6: BIBO Stability of Continuous Time Systems

Introduction

BIBO stability stands for bounded input, bounded output stability. BIBO stablity is the system property that any bounded input yields a bounded output. This is to say that as long as we input a signal with absolute value less than some constant, we are guaranteed to have an output with absolute value less than some other constant.

Continuous Time BIBO Stability

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, including the end behavior.

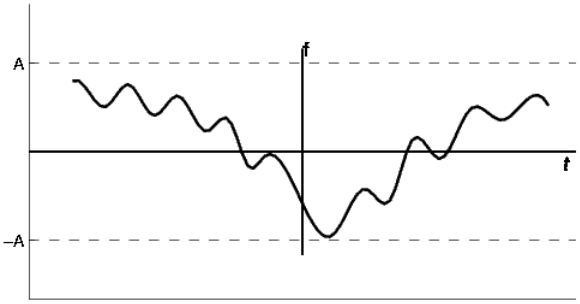


Figure 3.6.1: A bounded signal is a signal for which there exists a constant A such that $\forall t : (|f(t)| < A)$

Time Domain Conditions

Now that 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**

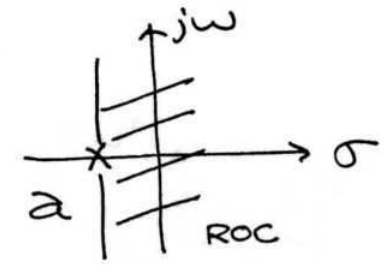
Continuous-Time Condition for BIBO Stability

$$\int_{-\infty}^{\infty} |h(t)| \mathrm{d}t < \infty$$

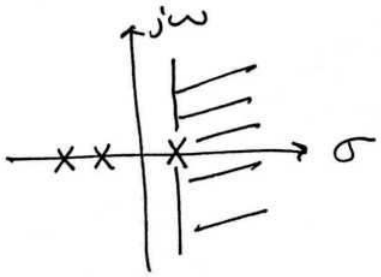
This is to say that the impulse response is **absolutely integrable**.

Laplace Domain Conditions

Stability is very easy to infer from the pole-zero plot of a transfer function. The only condition necessary to demonstrate stability is to show that the $i\omega$-axis is in the region of convergence. Consequently, for stable causal systems, all poles must be to the left of the imaginary axis.



(a)



(b)

Figure 3.6.2: (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.

BIBO Stability Summary

Bounded input bounded output stability, also known as BIBO stability, is an important and generally desirable system characteristic. A system is BIBO stable if every bounded input signal results in a bounded output signal, where boundedness is the property that the absolute value of a signal does not exceed some finite constant. In terms of time domain features, a continuous time system is BIBO stable if and only if its impulse response is absolutely integrable. Equivalently, in terms of Laplace domain features, a continuous time system is BIBO stable if and only if the region of convergence of the transfer function includes the imaginary axis.

This page titled 3.6: BIBO Stability of Continuous Time Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

3.7: Linear Constant Coefficient Differential Equations

Introduction: Ordinary Differential Equations

In our study of signals and systems, it will often be useful to describe systems using equations involving the rate of change in some quantity. Such equations are called differential equations. For instance, you may remember from a past physics course that an object experiences simple harmonic motion when it has an acceleration that is proportional to the magnitude of its displacement and opposite in direction. Thus, this system is described as the differential equation shown in Equation 3.7.1.

$$\frac{d^2x}{dt^2} = -cx \tag{3.7.1}$$

Because the differential equation in Equation 3.7.1 has only one independent variable and only has derivatives with respect to that variable, it is called an ordinary differential equation. There are more complicated differential equations, such as the Schrödinger equation, which involve derivatives with respect to multiple independent variables. These are called partial differential equations, but they are not within the scope of this module.

Given a sufficiently descriptive set of initial conditions or boundary conditions, if there is a solution to the differential equation, that solution is unique and describes the behavior of the system. Of course, the results are only accurate to the degree that the model mirrors reality.

Linear Constant Coefficient Ordinary Differential Equations

An important subclass of ordinary differential equations is the set of linear constant coefficient ordinary differential equations. These equations are of the form

$$Ax(t) = f(t) \tag{3.7.2}$$

where A is a differential operator of the form given in Equation 3.7.3.

$$A = a_n\frac{d^n}{dt^n} + a_{n-1}\frac{d^{n-1}}{dt^{n-1}} + ... + a_1\frac{d}{dt} + a_0 \tag{3.7.3}$$

Note that operators of this type satisfy the linearity conditions, and $a_1, ..., a_n$ are real constants. Furthermore, Equation 3.7.2 with these operators has derivatives with respect to only one variable, making it an ordinary differential equation.

A similar concept for a discrete time setting, difference equations, is discussed in the chapter on time domain analysis of discrete time systems. There are many parallels between the discussion of linear constant coefficient ordinary differential equations and linear constant coefficient difference equations.

Applications of Differential Equations

Consider the decay model in which a quantity of an unstable isotope decreases at a rate proportional to the quantity of unstable isotope remaining. Thus, the decay of the isotope is modeled by the first order linear constant coefficient differential equation

$$\frac{dx}{dt} + rx = 0$$

where r is some real rate.

Now consider the series RLC circuit shown in Figure 3.7.1. This system can be modeled using differential equations. We can use the voltage equations for each circuit element and Kirchoff's voltage law to write a second order linear constant coefficient differential equation describing the charge on the capacitor.

The voltage across the battery is simply V. The voltage across the capacitor is $\frac{1}{C}q$. The voltage across the resistor is $R\frac{dq}{dt}$. Finally, the voltage across the inductor is $L\frac{d^2q}{dt^2}$. Therefore, by Kirchoff's voltage law, it follows that

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q = V.$$

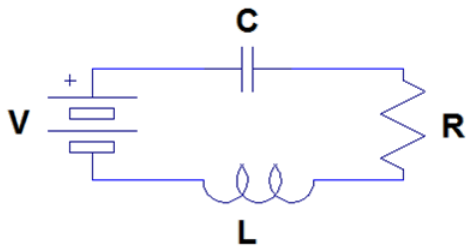


Figure 3.7.1: A series RLC circuit.

The section Solving Linear Constant Coefficient Differential Equations will describe in depth how solutions to differential equations like those in the examples may be obtained.

Linear Constant Coefficient Ordinary Differential Equations Summary

Differential equations are an important mathematical tool for modeling continuous time systems. An important subclass of these is the class of linear constant coefficient ordinary differential equations. Linear constant coefficient ordinary differential equations are often particularly easy to solve as will be described in the module on solutions to linear constant coefficient ordinary differential equations and are useful in describing a wide range of situations that arise in electrical engineering and in other fields.

This page titled 3.7: Linear Constant Coefficient Differential Equations is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

3.8: Solving Linear Constant Coefficient Differential Equations

Introduction

The approach to solving linear constant coefficient ordinary differential equations is to find the general form of all possible solutions to the equation and then apply a number of conditions to find the appropriate solution. The two main types of problems are initial value problems, which involve constraints on the solution and its derivatives at a single point, and boundary value problems, which involve constraints on the solution or its derivatives at several points.

The number of initial conditions needed for an Nth order differential equation, which is the order of the highest order derivative, is N, and a unique solution is always guaranteed if these are supplied. Boundary value problems can be slightly more complicated and will not necessarily have a unique solution or even a solution at all for a given set of conditions. Thus, this module will focus exclusively on initial value problems.

Solving Linear Constant Coefficient Ordinary Differential Equations

Consider some linear constant coefficient ordinary differential equation given by $Ax(t) = f(t)$, where A is a differential operator of the form

$$A = a_n \frac{d^n}{dt^n} + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} + ... + a_1 \frac{d}{dt} + a_0$$

Let $x_h(t)$ and $x_p(t)$ be two functions such that $Ax_h(t) = 0$ and $Ax_p(t) = f(t)$. By the linearity of A, note that $A(x_h(t) + x_p(t)) = 0 + f(t) = f(t)$. Thus, the form of the general solution $x_g(t)$ to any linear constant coefficient ordinary differential equation is the sum of a homogeneous solution $x_h(t)$ to the equation $Ax = 0$ and a particular solution $x_p(t)$ that is specific to the forcing function $f(t)$.

We wish to determine the forms of the homogeneous and nonhomogeneous solutions in full generality in order to avoid incorrectly restricting the form of the solution before applying any conditions. Otherwise, a valid set of initial or boundary conditions might appear to have no corresponding solution trajectory. The following discussion shows how to accomplish this for linear constant coefficient ordinary differential equations.

Finding the Homogeneous Solution

In order to find the homogeneous solution to $Ax(t) = f(t)$, consider the differential equation $Ax(t) = 0$. We know that the solutions have the form $ce^{\lambda t}$ for some complex constants c, λ. Since $Ace^{\lambda t} = 0$ for a solution, it follows that

$$\left(a_n \frac{d^n}{dt^n} + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} + ... + a_1 \frac{d}{dt} + a_0 \right) e^{\lambda t} = 0,$$

so it also follows that

$$a_n \lambda^n + a_{n-1} \lambda^{n-1} ... + a_1 \lambda + a_0 = 0.$$

Therefore, the parameters of the solution exponents are the roots of the above polynomial, called the characteristic polynomial.

For equations of order two or more, there will be several roots. If all of the roots are distinct, then the the general form of the homogeneous solution is simply

$$x_h(t) = c_1 e^{\lambda_1 t} + ... + c_n e^{\lambda_n t}.$$

If a root has multiplicity that is greater than one, the repeated solutions must be multiplied by each powers of t from 0 to one less than the root multiplicity (in order to ensure linearly independent solutions). For instance, if λ_1 had multiplicity 2 and λ_2 had multiplicity 3, the homogeneous solution would be

$$x_h(t) = c_1 e^{\lambda_1 t} + c_2 t e^{\lambda_1 t} + c_3 e^{\lambda_2 t} + c_4 t e^{\lambda_2 t} + c_5 t^2 e^{\lambda_2 t}$$

Example 3.8.1

Consider the decay model in which a quantity of an unstable isotope decreases at a rate proportional to the quanity of unstable isotope remaining. Thus, the decay of the isotope is modeled by the first order linear constant coefficient differential equation

$$\frac{dx}{dt} + rx = 0$$

where r is some real rate. This differential equation could easily be solved through straightforward integration. However, the methods described above will be used instead. Note that the forcing function is zero, so only a homogenous solution is needed. It is easy to see that the characteristic polynomial is $\lambda + r = 0$, so there is one root $\lambda_1 = r$. Thus the solution is of the form

$$x(t) = c_1 e^{rt}$$

Given a rate and an initial condition, this can be applied to a specific situation. For instance, we know that carbon-14 decays at a rate of approximately $r = 1.21 \times 10^{-4}$ year^{-1}, and if we normalize the natural concentration of carbon-14 to $x(0) = 1$ the solution becomes $x(t) = e^{-1.21 \times 10^{-4} t}$. Knowledge of this curve would be useful for radioisotope based dating.

Finding the Particular Solution

Finding the particular solution is slightly more complicated task than finding the homogeneous solution. A formal method, called variation of parameters accomplishes this, and there are also several heuristics that can be used. It can also be found through convolution of the input with the unit impulse response, once the unit impulse response is known. Finding the particular solution to a differential equation is discussed further in the chapter concerning the Laplace transform, which greatly simplifies the procedure for solving linear constant coefficient ordinary differential equations using frequency domain tools.

Example 3.8.2

Consider the series RLC circuit shown in Figure 3.8.1. This system can be modeled using differential equations. We can use the voltage equations for each circuit element and Kirchoff's voltage law to write a second order linear constant coefficient differential equation describing the charge on the capacitor.

The voltage across the battery is simply V. The voltage across the capacitor is $\frac{1}{C}q$. The voltage across the resistor is $R\frac{dq}{dt}$. Finally, the voltage across the inductor is $L\frac{d^2q}{dt^2}$. Therefore, by Kirchoff's voltage law, it follows that

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q = V$$

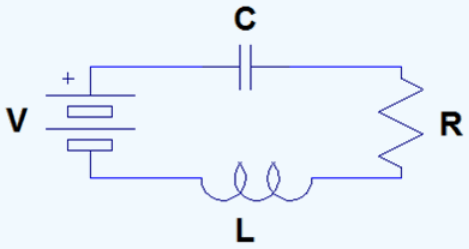


Figure 3.8.1: A series RLC circuit.

First, the homogeneous solution is found. It is easy to see that the characteristic polynomial is $L\lambda^2 + R\lambda + \frac{1}{C} = 0$. Therefore, the two roots are $\lambda_1 = \frac{-R-\sqrt{R^2-\frac{4L}{C}}}{2L}$ and $\lambda_2 = \frac{-R+\sqrt{R^2-\frac{4L}{C}}}{2L}$. Often, these are stated in terms of the attenuation factor $\alpha = \frac{R}{2}\sqrt{\frac{C}{L}}$ and the resonant frequency $\omega_0 = \frac{1}{\sqrt{LC}}$. Thus, $\lambda_1 = -\alpha - \sqrt{\alpha^2 - \omega_0^2}$ and $\lambda_2 = -\alpha + \sqrt{\alpha^2 - \omega_0^2}$.

Thus, the homogeneous equation is of the form

$$x_h(t) = c_1 e^{-\alpha - \sqrt{\alpha^2 - \omega_0^2}} + c_2 e^{-\alpha + \sqrt{\alpha^2 - \omega_0^2}}.$$

It turns out that the response to the constant voltage source forcing function is a constant, so

$$x_p(t) = VC.$$

Hence, the general solution is

$$x(t) = VC + c_1 e^{-\alpha - \sqrt{\alpha^2 - \omega_0^2}t} + c_2 e^{-\alpha + \sqrt{\alpha^2 - \omega_0^2}t}$$

where c_1 and c_2 depend on the initial conditions. The system demonstrates a rich array of behaviors based on the relative values of α and ω_0, which the reader is encouraged to explore.

Solving Differential Equations Summary

Linear constant coefficient ordinary differential equations are useful for modeling a wide variety of continuous time systems. The approach to solving them is to find the general form of all possible solutions to the equation and then apply a number of conditions to find the appropriate solution. This is done by finding the homogeneous solution to the differential equation that does not depend on the forcing function input and a particular solution to the differential equation that does depend on the forcing function input.

This page titled 3.8: Solving Linear Constant Coefficient Differential Equations is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

4: Time Domain Analysis of Discrete Time Systems

4.1: Discrete Time Systems
4.2: Discrete Time Impulse Response
4.3: Discrete Time Convolution
4.4: Properties of Discrete Time Convolution
4.5: Eigenfunctions of Discrete Time LTI Systems
4.6: BIBO Stability of Discrete Time Systems
4.7: Linear Constant Coefficient Difference Equations
4.8: Solving Linear Constant Coefficient Difference Equations

---

This page titled 4: Time Domain Analysis of Discrete Time Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

4.1: Discrete Time Systems

Introduction

As you already now know, a discrete time system operates on a discrete time signal input and produces a discrete time signal output. There are numerous examples of useful discrete time systems in digital signal processing, such as digital filters for images or sound. The class of discrete time systems that are both linear and time invariant, known as discrete time LTI systems, is of particular interest as the properties of linearity and time invariance together allow the use of some of the most important and powerful tools in signal processing.

Discrete Time Systems

Linearity and Time Invariance

A system H is said to be linear if it satisfies two important conditions. The first, additivity, states for every pair of signals x, y that $H(x+y) = H(x) + H(y)$. The second, homogeneity of degree one, states for every signal x and scalar a we have $H(ax) = aH(x)$. It is clear that these conditions can be combined together into a single condition for linearity. Thus, a system is said to be linear if for every signals x, y and scalars a, b we have that

$$H(ax + by) = aH(x) + bH(y).$$

Linearity is a particularly important property of systems as it allows us to leverage the powerful tools of linear algebra, such as bases, eigenvectors, and eigenvalues, in their study.

A system H is said to be time invariant if a time shift of an input produces the corresponding shifted output. In other, more precise words, the system H commutes with the time shift operator S_T for every $T \in \mathbb{Z}$. That is,

$$S_T H = H S_T.$$

Time invariance is desirable because it eases computation while mirroring our intuition that, all else equal, physical systems should react the same to identical inputs at different times.

When a system exhibits both of these important properties it opens. As will be explained and proven in subsequent modules, computation of the system output for a given input becomes a simple matter of convolving the input with the system's impulse response signal. Also proven later, the fact that complex exponential are eigenvectors of linear time invariant systems will encourage the use of frequency domain tools such as the various Fourier transforms and associated transfer functions, to describe the behavior of linear time invariant systems.

Example 4.1.1

Consider the system H in which

$$H(x[n]) = 2x[n]$$

for all signals f. Given any two signals f, g and scalars a, b

$$H(af[n] + bg[n])) = 2(af[n] + bg[n]) = a2f[n] + b2g[n] = aH(f[n]) + bH(g[n])$$

for all integers n. Thus, H is a linear system. For all integers T and signals x,

$$S_T(H(x[n])) = S_T(2x[n]) = 2x[n-T] = H(x[n-T]) = H\left(S_T(x[n])\right)$$

for all integers n. Thus, H is a time invariant system. Therefore, H is a linear time invariant system.

Difference Equation Representation

It is often useful to to describe systems using equations involving the rate of change in some quantity. For discrete time systems, such equations are called difference equations, a type of recurrence relation. One important class of difference equations is the set of linear constant coefficient difference equations, which are described in more detail in subsequent modules.

Example 4.1.2

Recall that the Fibonacci sequence describes a (very unrealistic) model of what happens when a pair rabbits get left alone in a black box... The assumptions are that a pair of rabbits never die and produce a pair of offspring every month starting on their second month of life. This system is defined by the recursion relation for the number of rabbit pairs $y[n]$ at month n

$$y[n] = y[n-1] + y[n-2]$$

with the initial conditions $y[0] = 0$ and $y[1] = 1$. The result is a very fast growth in the sequence. This is why we never leave black boxes open.

Discrete Time Systems Summary

Many useful discrete time systems will be encountered in a study of signals and systems. This course is most interested in those that demonstrate both the linearity property and the time invariance property, which together enable the use of some of the most powerful tools of signal processing. It is often useful to describe them in terms of rates of change through linear constant coefficient difference equations, a type of recurrence relation.

This page titled 4.1: Discrete Time Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

4.2: Discrete Time Impulse Response

Introduction

The output of a discrete time LTI system is completely determined by the input and the system's response to a unit impulse.

System Output

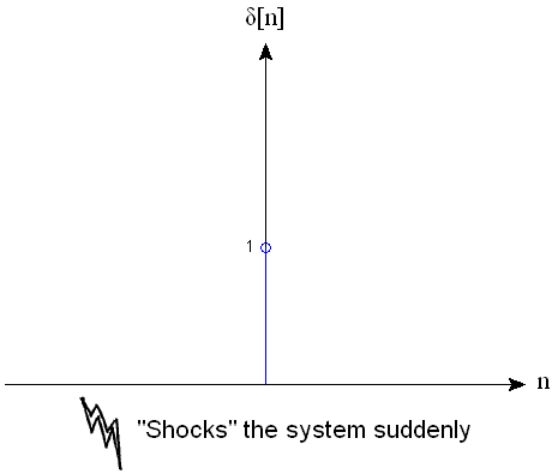
$$x[n] \rightarrow \boxed{\text{H}} \rightarrow y[n]$$

Figure 4.2.1: We can determine the system's output, $y[n]$, if we know the system's impulse response, $h[n]$, and the input, $x[n]$.

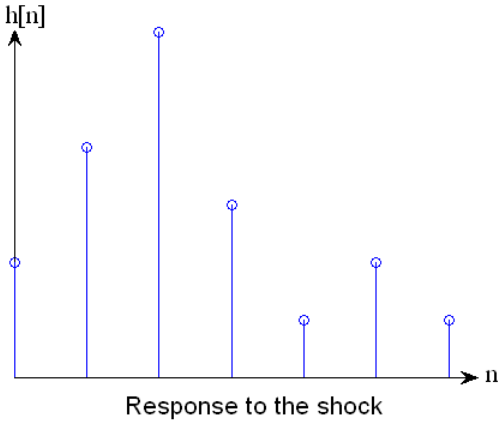
The output for a unit impulse input is called the impulse response.

$$\underset{\text{Impulse Input}}{\delta[n]} \rightarrow \boxed{\text{H}} \rightarrow \underset{\text{Impulse Response}}{h[n]}$$

Figure 4.2.2



(a)



(b)

Figure 4.2.3

Example Impulses

Since we are considering discrete time signals and systems, an ideal impulse is easy to simulate on a computer or some other digital device. It is simply a signal that is 1 at the point n = 0, and 0 everywhere else.

LTI Systems and Impulse Responses

Finding System Outputs

By the sifting property of impulses, any signal can be decomposed in terms of an infinite sum of shifted, scaled impulses.

$$\begin{aligned} x[n] &= \sum_{k=-\infty}^{\infty} x[k]\delta_k[n] \\ &= \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \end{aligned} \tag{4.2.1}$$

The function $\delta_k[\mathrm{n}] = \delta[\mathrm{n}-\mathrm{k}]$ peaks up where $n = k$.

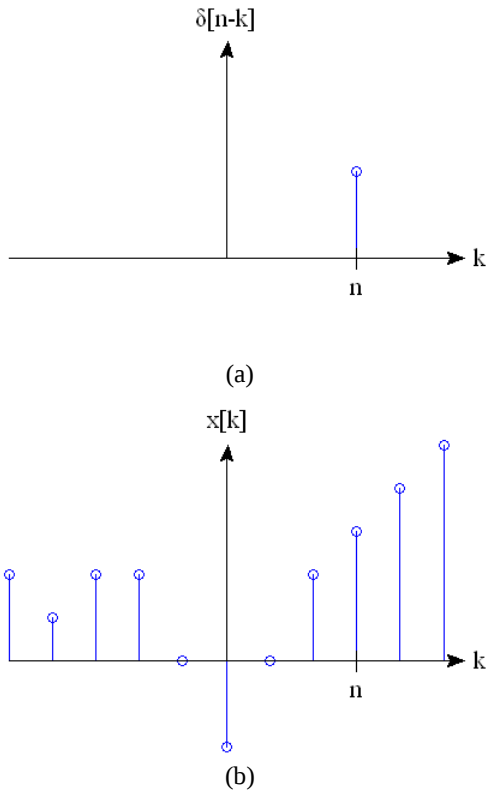


Figure 4.2.4

Since we know the response of the system to an impulse and any signal can be decomposed into impulses, all we need to do to find the response of the system to any signal is to decompose the signal into impulses, calculate the system's output for every impulse and add the outputs back together. This is the process known as **Convolution**. Since we are in **Discrete Time**, this is the **Discrete Time Convolution Sum**.

Finding Impulse Responses

Theory:

1. Solve the system's **Difference Equation** for y[n] with f[n] = δ[n]
2. Use the **Z-Transform**

Practice:

1. Apply an **impulse** input signal to the system and record the output
2. Use **Fourier methods**

We will assume that $h[n]$ is given for now. The goal is now to compute the output $y[n]$ given the impulse response $h[n]$ and the input $x[n]$.

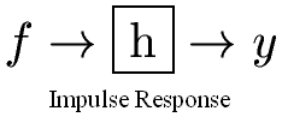


Figure 4.2.5

Impulse Response Summary

When a system is "shocked" by a delta function, it produces an output known as its impulse response. For an LTI system, the impulse response completely determines the output of the system given any arbitrary input. The output can be found using discrete time convolution.

This page titled 4.2: Discrete Time Impulse Response is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

4.3: Discrete Time Convolution

Introduction

Convolution, one of the most important concepts in electrical engineering, can be used to determine the output a system produces for a given input signal. It can be shown that a linear time invariant system is completely characterized by its impulse response. The sifting property of the discrete time impulse function tells us that the input signal to a system can be represented as a sum of scaled and shifted unit impulses. Thus, by linearity, it would seem reasonable to compute of the output signal as the sum of scaled and shifted unit impulse responses. That is exactly what the operation of convolution accomplishes. Hence, convolution can be used to determine a linear time invariant system's output from knowledge of the input and the impulse response.

Convolution and Circular Convolution

Convolution

Operation Definition

Discrete time convolution is an operation on two discrete time signals defined by the integral

$$(f * g)[n] = \sum_{k=-\infty}^{\infty} f[k]g[n-k]$$

for all signals f, g defined on $\mathbb{Z}$. It is important to note that the operation of convolution is commutative, meaning that

$$f * g = g * f$$

for all signals f, g defined on $\mathbb{Z}$. Thus, the convolution operation could have been just as easily stated using the equivalent definition

$$(f * g)[n] = \sum_{k=-\infty}^{\infty} f[n-k]g[k]$$

for all signals f, g defined on $\mathbb{Z}$. Convolution has several other important properties not listed here but explained and derived in a later module.

Definition Motivation

The above operation definition has been chosen to be particularly useful in the study of linear time invariant systems. In order to see this, consider a linear time invariant system H with unit impulse response h. Given a system input signal x we would like to compute the system output signal $H(x)$. First, we note that the input can be expressed as the convolution

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

by the sifting property of the unit impulse function. By linearity

$$H(x[n]) = \sum_{k=-\infty}^{\infty} x[k]H(\delta[n-k]).$$

Since $H(\delta[n-k])$ is the shifted unit impulse response $h[n-k]$, this gives the result

$$H(x[n]) = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = (x * h)[n].$$

Hence, convolution has been defined such that the output of a linear time invariant system is given by the convolution of the system input with the system unit impulse response.

Graphical Intuition

It is often helpful to be able to visualize the computation of a convolution in terms of graphical processes. Consider the convolution of two functions f, g given by

$$(f * g)[n] = \sum_{k=-\infty}^{\infty} f[k]g[n-k] = \sum_{k=-\infty}^{\infty} f[n-k]g[k].$$

The first step in graphically understanding the operation of convolution is to plot each of the functions. Next, one of the functions must be selected, and its plot reflected across the $k = 0$ axis. For each real n, that same function must be shifted left by n. The point-wise product of the two resulting plots is then computed, and then all of the values are summed.

Example 4.3.1

Recall that the impulse response for a discrete time echoing feedback system with gain a is

$$h[n] = a^n u[n],$$

and consider the response to an input signal that is another exponential

$$x[n] = b^n u[n].$$

We know that the output for this input is given by the convolution of the impulse response with the input signal

$$y[n] = x[n] * h[n].$$

We would like to compute this operation by beginning in a way that minimizes the algebraic complexity of the expression. However, in this case, each possible choice is equally simple. Thus, we would like to compute

$$y[n] = \sum_{k=-\infty}^{\infty} a^k u[k] b^{n-k} u[n-k].$$

The step functions can be used to further simplify this sum. Therefore,

$$y[n] = 0$$

for $n < 0$ and

$$y[n] = \sum_{k=0}^{n} [ab]^k$$

for $n \geq 0$. Hence, provided $ab \neq 1$, we have that

$$y[n] = \begin{cases} 0 & n < 0 \\ \frac{1-(ab)^{n+1}}{1-(ab)} & n \geq 0 \end{cases}$$

Circular Convolution

Discrete time circular convolution is an operation on two finite length or periodic discrete time signals defined by the sum

$$(f \circledast g)[n] = \sum_{k=0}^{N-1} \hat{f}[k]\hat{g}[n-k]$$

for all signals f, g defined on $\mathbb{Z}[0, N-1]$ where $\hat{f}$, $\hat{g}$ are periodic extensions of f and g. It is important to note that the operation of circular convolution is commutative, meaning that

$$f \circledast g = g \circledast f$$

for all signals f, g defined on $\mathbb{Z}[0, N-1]$. Thus, the circular convolution operation could have been just as easily stated using the equivalent definition

$$(f \circledast g)[n] = \sum_{k=0}^{N-1} \hat{f}[n-k]\hat{g}[k]$$

for all signals f, g defined on $\mathbb{Z}[0, N-1]$ where $\hat{f}$, $\hat{g}$ are periodic extensions of f and g. Circular convolution has several other important properties not listed here but explained and derived in a later module.

Alternatively, discrete time circular convolution can be expressed as the sum of two summations given by

$$(f \circledast g)[n] = \sum_{k=0}^{n} f[k]g[n-k] + \sum_{k=n+1}^{N-1} f[k]g[n-k+N]$$

for all signals f, g defined on $\mathbb{Z}[0, N-1]$.

Meaningful examples of computing discrete time circular convolutions in the time domain would involve complicated algebraic manipulations dealing with the wrap around behavior, which would ultimately be more confusing than helpful. Thus, none will be provided in this section. Of course, example computations in the time domain are easy to program and demonstrate. However, discrete time circular convolutions are more easily computed using frequency domain tools as will be shown in the discrete time Fourier series section.

Definition Motivation

The above operation definition has been chosen to be particularly useful in the study of linear time invariant systems. In order to see this, consider a linear time invariant system H with unit impulse response h. Given a periodic system input signal x we would like to compute the system output signal $H(x)$. First, we note that the input can be expressed as the circular convolution

$$x[n] = \sum_{k=0}^{N-1} \hat{x}[k]\hat{\delta}[n-k]$$

by the sifting property of the unit impulse function. By linearity,

$$H(x[n]) = \sum_{k=0}^{N-1} \hat{x}[k]H(\hat{\delta}[n-k]).$$

Since $H(\delta[n-k])$ is the shifted unit impulse response $h[n-k]$, this gives the result

$$H(x[n]) = \sum_{k=0}^{N-1} \hat{x}[k]\hat{h}[n-k] = (x \circledast h)[n].$$

Hence, circular convolution has been defined such that the output of a linear time invariant system is given by the convolution of the system input with the system unit impulse response.

Graphical Intuition

It is often helpful to be able to visualize the computation of a circular convolution in terms of graphical processes. Consider the circular convolution of two finite length functions f, g given by

$$(f \circledast g)[n] = \sum_{k=0}^{N-1} \hat{f}[k]\hat{g}[n-k] = \sum_{k=0}^{N-1} \hat{f}[n-k]\hat{g}[k]$$

The first step in graphically understanding the operation of convolution is to plot each of the periodic extensions of the functions. Next, one of the functions must be selected, and its plot reflected across the $k = 0$ axis. For each $n \in \mathbb{Z}[0, N-1]$, that same function must be shifted left by n. The point-wise product of the two resulting plots is then computed, and finally all of these values are summed.

Interactive Element

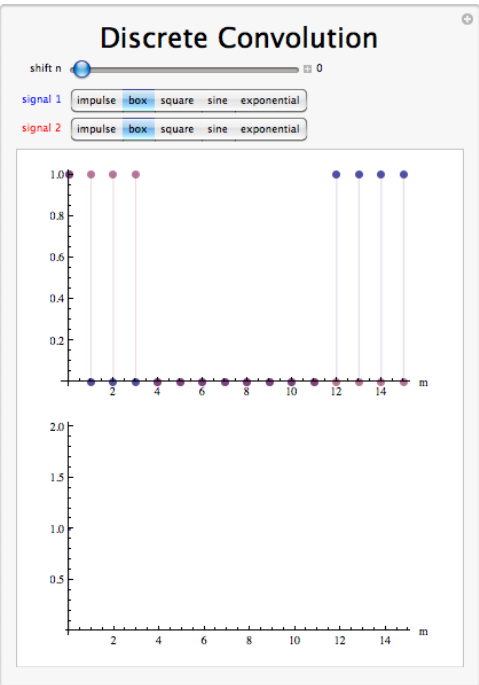


Figure 4.3.1: Interact (when online) with the Mathematica CDF demonstrating Discrete Linear Convolution. To download, right click and save file as .cdf

Convolution Summary

Convolution, one of the most important concepts in electrical engineering, can be used to determine the output signal of a linear time invariant system for a given input signal with knowledge of the system's unit impulse response. The operation of discrete time convolution is defined such that it performs this function for infinite length discrete time signals and systems. The operation of

discrete time circular convolution is defined such that it performs this function for finite length and periodic discrete time signals. In each case, the output of the system is the convolution or circular convolution of the input signal with the unit impulse response.

This page titled 4.3: Discrete Time Convolution is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

4.4: Properties of Discrete Time Convolution

Introduction

We have already shown the important role that discrete time convolution plays in signal processing. This section provides discussion and proof of some of the important properties of discrete time convolution. Analogous properties can be shown for discrete time circular convolution with trivial modification of the proofs provided except where explicitly noted otherwise.

Discrete Time Convolution Properties

Associativity

The operation of convolution is associative. That is, for all discrete time signals f_1, f_2, f_3 the following relationship holds.

$$f_1 * (f_2 * f_3) = (f_1 * f_2) * f_3$$

In order to show this, note that

$$\begin{aligned}
(f_1 * (f_2 * f_3))[n] &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f_1[k_1] f_2[k_2] f_3[(n-k_1)-k_2] \\
&= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f_1[k_1] f_2[(k_1+k_2)-k_1] f_3[n-(k_1+k_2)] \\
&= \sum_{k_3=-\infty}^{\infty} \sum_{k_1=-\infty}^{\infty} f_1[k_1] f_2[k_3-k_1] f_3[n-k_3] \\
&= ((f_1 * f_2) * f_3)[n]
\end{aligned} \tag{4.4.1}$$

proving the relationship as desired through the substitution $k_3 = k_1 + k_2$.

Commutativity

The operation of convolution is commutative. That is, for all discrete time signals f_1, f_2 the following relationship holds.

$$f_1 * f_2 = f_2 * f_1$$

In order to show this, note that

$$\begin{aligned}
(f_1 * f_2)[n] &= \sum_{k_1=-\infty}^{\infty} f_1[k_1] f_2[n-k_1] \\
&= \sum_{k_2=-\infty}^{\infty} f_1[n-k_2] f_2[k_2] \\
&= (f_2 * f_1)[n]
\end{aligned} \tag{4.4.2}$$

proving the relationship as desired through the substitution $k_2 = n - k_1$.

Distribitivity

The operation of convolution is distributive over the operation of addition. That is, for all discrete time signals f_1, f_2, f_3 the following relationship holds.

$$f_1 * (f_2 + f_3) = f_1 * f_2 + f_1 * f_3$$

In order to show this, note that

$$\begin{aligned}
(f_1 * (f_2 + f_3))(n) &= \sum_{k=-\infty}^{\infty} f_1(k)(f_2(n-k) + f_3(n-k)) \\
&= \sum_{k=-\infty}^{\infty} f_1(k) f_2(n-k) + \sum_{k=-\infty}^{\infty} f_1(k) f_3(n-k) \\
&= (f_1 * f_2 + f_1 * f_3)(n)
\end{aligned} \tag{4.4.3}$$

proving the relationship as desired.

Multilinearity

The operation of convolution is linear in each of the two function variables. Additivity in each variable results from distributivity of convolution over addition. Homogenity of order one in each variable results from the fact that for all discrete time signals f_1, f_2 and scalars aa the following relationship holds.

$$a\left(f_1 * f_2\right) = \left(a f_1\right) * f_2 = f_1 * \left(a f_2\right)$$

In order to show this, note that

$$\begin{aligned}
\left(a\left(f_1 * f_2\right)\right)[n] &= a \sum_{k=-\infty}^{\infty} f_1[k] f_2[n-k] \\
&= \sum_{k=-\infty}^{\infty} \left(a f_1[k]\right) f_2[n-k] \\
&= \left(\left(a f_1\right) * f_2\right)[n] \\
&= \sum_{k=-\infty}^{\infty} f_1[k]\left(a f_2[n-k]\right) \\
&= \left(f_1 * \left(a f_2\right)\right)[n]
\end{aligned} \tag{4.4.4}$$

proving the relationship as desired.

Conjugation

The operation of convolution has the following property for all discrete time signals f_1, f_2.

$$\overline{f_1^* f_2} = \overline{f_1} * \overline{f_2}$$

In order to show this, note that

$$\begin{aligned}
\overline{(f_1 * f_2)}[n] &= \overline{\sum_{k=-\infty}^{\infty} f_1[k] f_2[n-k]} \\
&= \sum_{k=-\infty}^{\infty} \overline{f_1[k] f_2[n-k]} \\
&= \sum_{k=-\infty}^{\infty} \overline{f_1}[k] \overline{f_2}[n-k] \\
&= (\overline{f_1} * \overline{f_2})[n]
\end{aligned} \tag{4.4.5}$$

proving the relationship as desired.

Time Shift

The operation of convolution has the following property for all discrete time signals f_1, f_2 where S_T is the time shift operator with $T \in \mathbb{Z}$.

$$S_T\left(f_1 * f_2\right) = \left(S_T f_1\right) * f_2 = f_1 * \left(S_T f_2\right)$$

In order to show this, note that

$$\begin{aligned}
S_T\left(f_1 * f_2\right)[n] &= \sum_{k=-\infty}^{\infty} f_2[k] f_1[(n-T)-k] \\
&= \sum_{k=-\infty}^{\infty} f_2[k] S_T f_1[n-k] \\
&= \left(\left(S_T f_1\right) * f_2\right)[n] \\
&= \sum_{k=-\infty}^{\infty} f_1[k] f_2[(n-T)-k] \\
&= \sum_{k=-\infty}^{\infty} f_1[k] S_T f_2[n-k] \\
&= f_1 * \left(S_T f_2\right)[n]
\end{aligned} \tag{4.4.6}$$

proving the relationship as desired.

Impulse Convolution

The operation of convolution has the following property for all discrete time signals f where δ is the unit sample function.

$$f * \delta = f$$

In order to show this, note that

$$\begin{aligned}(f * \delta)[n] &= \sum_{k=-\infty}^{\infty} f[k]\delta[n-k] \\ &= f[n] \sum_{k=-\infty}^{\infty} \delta[n-k] \\ &= f[n]\end{aligned} \tag{4.4.7}$$

proving the relationship as desired.

Width

The operation of convolution has the following property for all discrete time signals f_1, f_2 where Duration(f) gives the duration of a signal f.

$$\text{Duration}\,(f_1 * f_2) = \text{Duration}\,(f_1) + \text{Duration}\,(f_2) - 1$$

In order to show this informally, note that $(f_1 * f_2)[n]$ is nonzero for all n for which there is a k such that $f_1[k]f_2[n-k]$ is nonzero. When viewing one function as reversed and sliding past the other, it is easy to see that such a k exists for all n on an interval of length Duration(f_1) + Duration(f_2) − 1. Note that this is not always true of circular convolution of finite length and periodic signals as there is then a maximum possible duration within a period.

Convolution Properties Summary

As can be seen the operation of discrete time convolution has several important properties that have been listed and proven in this module. With silight modifications to proofs, most of these also extend to discrete time circular convolution as well and the cases in which exceptions occur have been noted above. These identities will be useful to keep in mind as the reader continues to study signals and systems.

This page titled 4.4: Properties of Discrete Time Convolution is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

4.5: Eigenfunctions of Discrete Time LTI Systems

Introduction

Prior to reading this module, the reader should already have some experience with linear algebra and should specifically be familiar with the eigenvectors and eigenvalues of linear operators. A linear time invariant system is a linear operator defined on a function space that commutes with every time shift operator on that function space. Thus, we can also consider the eigenvector functions, or eigenfunctions, of a system. It is particularly easy to calculate the output of a system when an eigenfunction is the input as the output is simply the eigenfunction scaled by the associated eigenvalue. As will be shown, discrete time complex exponentials serve as eigenfunctions of linear time invariant systems operating on discrete time signals.

Eigenfunctions of LTI Systems

Consider a linear time invariant system H with impulse response hh operating on some space of infinite length discrete time signals. Recall that the output $H(x[n])$ of the system for a given input $x[n]$ is given by the discrete time convolution of the impulse response with the input

$$H(x[n]) = \sum_{k=-\infty}^{\infty} h[k]x[n-k].$$

Now consider the input $x(n) = e^{sn}$ where $s \in \mathbb{C}$. Computing the output for this input,

$$\begin{aligned} H\left(e^{sn}\right) &= \sum_{k=-\infty}^{\infty} h[k]e^{s(n-k)} \\ &= \sum_{k=-\infty}^{\infty} h[k]e^{sn}e^{-sk} \\ &= e^{sn}\sum_{k=-\infty}^{\infty} h[k]e^{-sk} \end{aligned} \quad (4.5.1)$$

Thus,

$$H\left(e^{sn}\right) = \lambda_s e^{sn}$$

where

$$\lambda_s = \sum_{k=-\infty}^{\infty} h[k]e^{-sk}$$

is the eigenvalue corresponding to the eigenvector e^{sn}.

There are some additional points that should be mentioned. Note that, there still may be additional eigenvalues of a linear time invariant system not described by e^{sn} for some $s \in \mathbb{C}$. Furthermore, the above discussion has been somewhat formally loose as e^{sn} may or may not belong to the space on which the system operates. However, for our purposes, complex exponentials will be accepted as eigenvectors of linear time invariant systems. A similar argument using discrete time circular convolution would also hold for spaces finite length signals.

Eigenfunction of LTI Systems Summary

As has been shown, discrete time complex exponential are eigenfunctions of linear time invariant systems operating on discrete time signals. Thus, it is particularly simple to calculate the output of a linear time invariant system for a complex exponential input as the result is a complex exponential output scaled by the associated eigenvalue. Consequently, representations of discrete time signals in terms of discrete time complex exponentials provide an advantage when studying signals. As will be explained later, this is what is accomplished by the discrete time Fourier transform and discrete time Fourier series, which apply to aperiodic and periodic signals respectively.

This page titled 4.5: Eigenfunctions of Discrete Time LTI Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

4.6: BIBO Stability of Discrete Time Systems

Introduction

BIBO stability stands for bounded input, bounded output stability. BIBO stability is the system property that any bounded input yields a bounded output. This is to say that as long as we input a signal with absolute value less than some constant, we are guaranteed to have an output with absolute value less than some other constant.

Discrete Time BIBO Stability

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, including the end behavior.

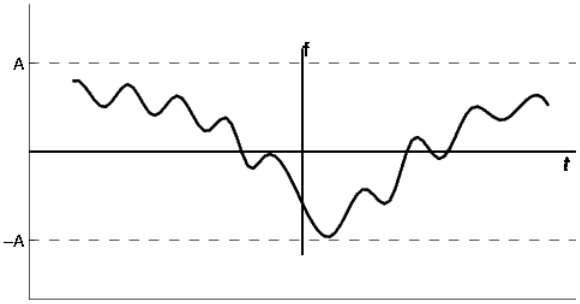


Figure 4.6.1: A bounded signal is a signal for which there exists a constant A such that $\forall n : (|f[n]| < A)$

Time Domain Conditions

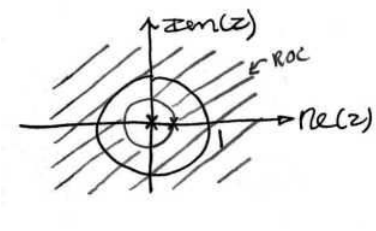
Now that we have identified what it means for a signal to be bounded, we must turn our attention to the condition a system must possess in order to guarantee that if any bounded signal is passed through the system, a bounded signal will arise on the output. It turns out that a continuous-time LTI (Section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

Discrete-Time Condition for BIBO Stability

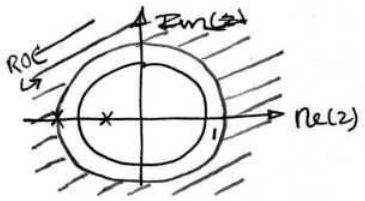
$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty$$

Z-Domain Conditions

Stability for discrete-time signals (Section 1.1) in the z-domain 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. Consequently, for stable causal systems, all poles must be within the unit circle.



(a)



(b)

Figure 4.6.2: (a) A stable discrete-time system. (b) An unstable discrete-time system.

BIBO Stability Summary

Bounded input bounded output stability, also known as BIBO stability, is an important and generally desirable system characteristic. A system is BIBO stable if every bounded input signal results in a bounded output signal, where boundedness is the property that the absolute value of a signal does not exceed some finite constant. In terms of time domain features, a discrete time system is BIBO stable if and only if its impulse response is absolutely summable. Equivalently, in terms of z-domain features, a continuous time system is BIBO stable if and only if the region of convergence of the transfer function includes the unit circle.

This page titled 4.6: BIBO Stability of Discrete Time Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

4.7: Linear Constant Coefficient Difference Equations

Introduction: Difference Equations

In our study of signals and systems, it will often be useful to describe systems using equations involving the rate of change in some quantity. In discrete time, this is modeled through difference equations, which are a specific type of recurrance relation. For instance, recall that the funds in an account with discretely compounded interest rate r will increase by r times the previous balance. Thus, a discretely compounded interest system is described by the first order difference equation shown in Equation 4.7.1.

$$y(n) = (1 + r)y(n - 1) \tag{4.7.1}$$

Given a sufficiently descriptive set of initial conditions or boundary conditions, if there is a solution to the difference equation, that solution is unique and describes the behavior of the system. Of course, the results are only accurate to the degree that the model mirrors reality.

Linear Constant Coefficient Difference Equations

An important subclass of difference equations is the set of linear constant coefficient difference equations. These equations are of the form

$$Cy(n) = f(n) \tag{4.7.2}$$

where C is a difference operator of the form given

$$C = c_N D^N + c_{N-1} D^{N-1} + ... + c_1 D + c_0 \tag{4.7.3}$$

in which D is the first difference operator

$$D(y(n)) = y(n) - y(n - 1). \tag{4.7.4}$$

Note that operators of this type satisfy the linearity conditions, and $c_0, \ldots, c_n$ are real constants.

However, Equation 4.7.2 can easily be written as a linear constant coefficient recurrence equation without difference operators. Conversely, linear constant coefficient recurrence equations can also be written in the form of a difference equation, so the two types of equations are different representations of the same relationship. Although we will still call them linear constant coefficient difference equations in this course, we typically will not write them using difference operators. Instead, we will write them in the simpler recurrence relation form

$$\sum_{k=0}^{N} a_k y(n - k) = \sum_{k=0}^{M} b_k x(n - k) \tag{4.7.5}$$

where x is the input to the system and y is the output. This can be rearranged to find $y(n)$ as

$$y(n) = \frac{1}{a_0} \left(- \sum_{k=1}^{N} a_k y(n - k) + \sum_{k=0}^{M} b_k x(n - k) \right) \tag{4.7.6}$$

The forms provided by Equation 4.7.5 and Equation 4.7.6 will be used in the remainder of this course.

A similar concept for continuous time setting, differential equations, is discussed in the chapter on time domain analysis of continuous time systems. There are many parallels between the discussion of linear constant coefficient ordinary differential equations and linear constant coefficient difference equations.

Applications of Difference Equations

Difference equations can be used to describe many useful digital filters as described in the chapter discussing the z-transform. An additional illustrative example is provided here.

Example 4.7.1

Recall that the Fibonacci sequence describes a (very unrealistic) model of what happens when a pair rabbits get left alone in a black box... The assumptions are that a pair of rabbits never die and produce a pair of offspring every month starting on their second month of life. This system is defined by the recursion relation for the number of rabbit pairs $y(n)$ at month n

$$y(n) = y(n-1) + y(n-2)$$

with the initial conditions $y(0) = 0$ and $y(1) = 1$. The result is a very fast growth in the sequence. This is why we do not open black boxes.

Linear Constant Coefficient Difference Equations Summary

Difference equations are an important mathematical tool for modeling discrete time systems. An important subclass of these is the class of linear constant coefficient difference equations. Linear constant coefficient difference equations are often particularly easy to solve as will be described in the module on solutions to linear constant coefficient difference equations and are useful in describing a wide range of situations that arise in electrical engineering and in other fields.

This page titled 4.7: Linear Constant Coefficient Difference Equations is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

4.8: Solving Linear Constant Coefficient Difference Equations

Introduction

The approach to solving linear constant coefficient difference equations is to find the general form of all possible solutions to the equation and then apply a number of conditions to find the appropriate solution. The two main types of problems are **initial value problems**, which involve constraints on the solution at several consecutive points, and **boundary value problems**, which involve constraints on the solution at nonconsecutive points.

The number of initial conditions needed for an Nth order difference equation, which is the order of the highest order difference or the largest delay parameter of the output in the equation, is N, and a unique solution is always guaranteed if these are supplied. Boundary value problems can be slightly more complicated and will not necessarily have a unique solution or even a solution at all for a given set of conditions. Thus, this section will focus exclusively on initial value problems.

Solving Linear Constant Coefficient Difference Equations

Consider some linear constant coefficient difference equation given by $Ay(n) = f(n)$, in which A is a difference operator of the form

$$A = a_N D^N + a_{N-1} D^{N-1} + ... + a_1 D + a_0$$

where D is the first difference operator

$$D(y(n)) = y(n) - y(n-1).$$

Let $y_h(n)$ and $y_p(n)$ be two functions such that $Ay_h(n) = 0$ and $Ay_p(n) = f(n)$. By the linearity of A, note that $L(y_h(n) + y_p(n)) = 0 + f(n) = f(n)$. Thus, the form of the general solution $y_g(n)$ to any linear constant coefficient ordinary differential equation is the sum of a homogeneous solution $y_h(n)$ to the equation $Ay(n) = 0$ and a particular solution $y_p(n)$ that is specific to the forcing function $f(n)$.

We wish to determine the forms of the homogeneous and nonhomogeneous solutions in full generality in order to avoid incorrectly restricting the form of the solution before applying any conditions. Otherwise, a valid set of initial or boundary conditions might appear to have no corresponding solution trajectory. The following sections discuss how to accomplish this for linear constant coefficient difference equations.

Finding the Homogeneous Solution

In order to find the homogeneous solution to a difference equation described by the recurrence relation

$$\sum_{k=0}^{N} a_k y(n-k) = f(n),$$

consider the difference equation

$$\sum_{k=0}^{N} a_k y(n-k) = 0.$$

We know that the solutions have the form $c\lambda^n$ for some complex constants c, λ. Since $\sum_{k=0}^{N} a_k c\lambda^{n-k} = 0$ for a solution it follows that

$$c\lambda^{n-N} \sum_{k=0}^{N} a_k \lambda^{N-k} = 0$$

so it also follows that

$$a_0 \lambda^N + ... + a_N = 0.$$

Therefore, the solution exponential are the roots of the above polynomial, called the characteristic polynomial.

For equations of order two or more, there will be several roots. If all of the roots are distinct, then the general form of the homogeneous solution is simply

$$y_h(n) = c_1\lambda_1^n + ... + c_2\lambda_2^n.$$

If a root has multiplicity that is greater than one, the repeated solutions must be multiplied by each power of n from 0 to one less than the root multiplicity (in order to ensure linearly independent solutions). For instance, if λ_1 had a multiplicity of 2 and λ_2 had multiplicity 3, the homogeneous solution would be

$$y_h(n) = c_1\lambda_1^n + c_2 n\lambda_1^n + c_3\lambda_2^n + c_4 n\lambda_2^n + c_5 n^2\lambda_2^n$$

Example 4.8.1: Fibonacci Sequence

Recall that the Fibonacci sequence describes a (very unrealistic) model of what happens when a pair rabbits get left alone in a black box... The assumptions are that a pair of rabits never die and produce a pair of offspring every month starting on their second month of life. This system is defined by the recursion relation for the number of rabit pairs $y(n)$ at month n

$$y(n) - y(n-1) - y(n-2) = 0$$

with the initial conditions $y(0) = 0$ and $y(1) = 1$.

Note that the forcing function is zero, so only the homogenous solution is needed. It is easy to see that the characteristic polynomial is $\lambda^2 - \lambda - 1 = 0$, so there are two roots with multiplicity one. These are $\lambda_1 = \frac{1+\sqrt{5}}{2}$ and $\lambda_2 = \frac{1-\sqrt{5}}{2}$. Thus, the solution is of the form

$$y(n) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Using the initial conditions, we determine that

$$c_1 = \frac{\sqrt{5}}{5}$$

and

$$c_2 = -\frac{\sqrt{5}}{5}.$$

Hence, the Fibonacci sequence is given by

$$y(n) = \frac{\sqrt{5}}{5}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{\sqrt{5}}{5}\left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Finding the Particular Solution

Finding the particular solution is a slightly more complicated task than finding the homogeneous solution. It can be found through convolution of the input with the unit impulse response once the unit impulse response is known. Finding the particular solution ot a differential equation is discussed further in the chapter concerning the z-transform, which greatly simplifies the procedure for solving linear constant coefficient differential equations using frequency domain tools.

Example 4.8.2

Consider the following difference equation describing a system with feedback

$$y(n) - ay(n-1) = x(n).$$

In order to find the homogeneous solution, consider the difference equation

$$y(n) - ay(n-1) = 0.$$

It is easy to see that the characteristic polynomial is $\lambda - a = 0$, so $\lambda = a$ is the only root. Thus the homogeneous solution is of the form

$$y_h(n) = c_1 a^n.$$

In order to find the particular solution, consider the output for the $x(n) = \delta(n)$ unit impulse case

$$y(n) - ay(n-1) = \delta(n).$$

By inspection, it is clear that the impulse response is $a^n u(n)$. Hence, the particular solution for a given $x(n)$ is

$$y_p(n) = x(n) * (a^n u(n)) .$$

Therefore, the general solution is

$$y_g(n) = y_h(n) + y_p(n) = c_1 a^n + x(n) * (a^n u(n)) .$$

Initial conditions and a specific input can further tailor this solution to a specific situation.

Solving Difference Equations Summary

Linear constant coefficient difference equations are useful for modeling a wide variety of discrete time systems. The approach to solving them is to find the general form of all possible solutions to the equation and then apply a number of conditions to find the appropriate solution. This is done by finding the homogeneous solution to the difference equation that does not depend on the forcing function input and a particular solution to the difference equation that does depend on the forcing function input.

This page titled 4.8: Solving Linear Constant Coefficient Difference Equations is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

5: Introduction to Fourier Analysis

5.1: Introduction to Fourier Analysis

This page titled 5: Introduction to Fourier Analysis is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

5.1: Introduction to Fourier Analysis

Fourier's Daring Leap

Fourier postulated around 1807 that any periodic signal (equivalently finite length signal) can be built up as an infinite linear combination of harmonic sinusoidal waves.

i.e. Given the collection

$$B = \left\{ e^{j\frac{2\pi}{T}nt} \right\}_{n=-\infty}^{\infty}$$

any

$$f(t) \in L^2[0,T)$$

can be approximated arbitrarily closely by

$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{j\frac{2\pi}{T}nt}.$$

Now, The issue of exact convergence did bring Fourier much criticism from the French Academy of Science (Laplace, Lagrange, Monge and LaCroix 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 practical viewpoint. See more in the section on Gibbs Phenomena.

Fourier analysis is fundamental to understanding the behavior of signals and systems. This is a result of the fact that sinusoids are Eigenfunctions (Section 14.5) of linear, time-invariant (LTI) (Section 2.2) 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 function) 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 3.4) (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 Series, Continuous-Time Fourier Transform (Section 8.2), Discrete-Time Fourier Transform (Section 9.2), and Discrete Fourier Transform. For this document, we will view the Laplace Transform (Section 11.1) and Z-Transform 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.

Table 5.1.1: 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

This page titled 5.1: Introduction to Fourier Analysis is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

6: Continuous Time Fourier Series (CTFS)

6.1: Continuous Time Periodic Signals
6.2: Continuous Time Fourier Series (CTFS)
6.3: Common Fourier Series
6.4: Properties of the CTFS
6.5: Continuous Time Circular Convolution and the CTFS
6.6: Convergence of Fourier Series
6.7: Gibbs Phenomena

This page titled 6: Continuous Time Fourier Series (CTFS) is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

6.1: Continuous Time Periodic Signals

Introduction

This module describes the type of signals acted on by the Continuous Time Fourier Series.

Relevant Spaces

The Continuous-Time Fourier Series maps finite-length (or T-periodic), continuous-time signals in L^2 to infinite-length, discrete-frequency signals in l^2.

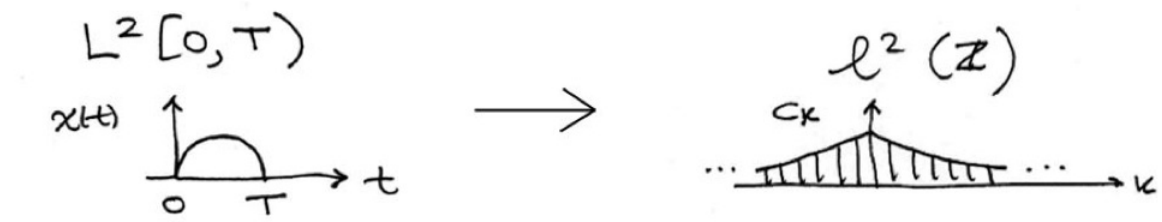


Figure 1: Mapping $L^2([0, T))$ in the time domain to $l^2(\mathbb{Z})$ in the frequency domain.

Figure 6.1.1: Mapping $L^2([0,T))$ in the time domain to $l^2(\mathbb{Z})$ in the frequency domain.

Periodic Signals

When a function repeats itself exactly after some given period, or cycle, we say it's **periodic**. A **periodic function** can be mathematically defined as:

$$f(t) = f(t + mT) \forall m : (m \in \mathbb{Z}) \tag{6.1.1}$$

where $T > 0$ represents the **fundamental period** of the signal, which is the smallest positive value of T for the signal to repeat. Because of this, you may also see a signal referred to as a T-periodic signal. Any function that satisfies this equation is said to be **periodic** with period T.

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

1. as functions on **all** of $\mathbb{R}$

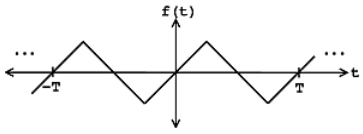


Figure 6.1.1: Continuous time periodic function over all of $\mathbb{R}$ where $f(t_0) = f(t_0 + T)$

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.

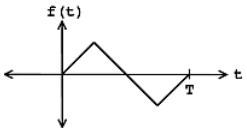


Figure 6.1.2: Remove the redundancy of the period function so that $f(t)$ is undefined outside $[0, T]$.

An **aperiodic** CT function $f(t)$, on the other hand, does not repeat for **any** $T \in \mathbb{R}$; i.e. there exists no T such that Equation 6.1.1 holds.

Demonstration

Here's an example demonstrating a **periodic** sinusoidal signal with various frequencies, amplitudes and phase delays:

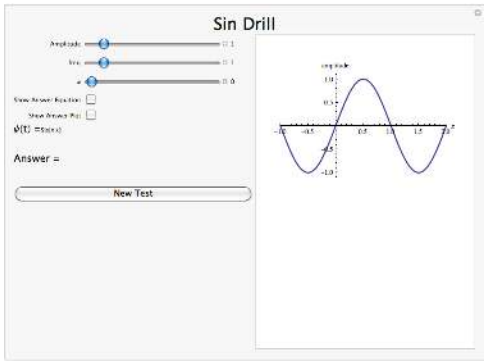


Figure 6.1.3: Interact (when online) with a Mathematica CDF demonstrating a Periodic Sinusoidal Signal with various frequencies, amplitudes, and phase delays. To download, right click and save file as .cdf.

To learn the full concept behind periodicity, see the video below.

Khan Lecture on Periodic Signals

video

from

Khan Academy

Conclusion

A periodic signal is completely defined by its values in one period, such as the interval $[0, T]$.

This page titled 6.1: Continuous Time Periodic Signals is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

6.2: Continuous Time Fourier Series (CTFS)

Introduction

In this module, we will derive an expansion for continuous-time, periodic functions, and in doing so, derive the **Continuous Time Fourier Series** (CTFS).

Since complex exponentials (Section 1.8) are eigenfunctions of linear time-invariant (LTI) systems (Section 14.5), calculating the output of an LTI system $\mathscr{H}$ given e^{st} as an input amounts to simple multiplication, where $H(s) \in \mathbb{C}$ is the eigenvalue corresponding to s. As shown in the figure, a simple exponential input would yield the output

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

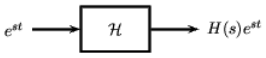


Figure 6.2.1: Simple LTI system.

Using this and the fact that $\mathscr{H}$ is linear, calculating $y(t)$ for combinations of complex exponentials is also straightforward.

$$c_1 e^{s_1 t} + c_2 e^{s_2 t} \to c_1 H(s_1) e^{s_1 t} + c_2 H(s_2) e^{s_2 t} \tag{6.2.1}$$

$$\sum_n c_n e^{s_n t} \to \sum_n c_n H(s_n) e^{s_n t} \tag{6.2.2}$$

The action of H on an input such as those in the two equations above is easy to explain. $\mathscr{H}$ **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 a system.

Fourier Series Synthesis

Joseph Fourier demonstrated that an arbitrary $f(t)$ can be written as a linear combination of harmonic complex sinusoids

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\omega_0 n t} \tag{6.2.3}$$

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 Equation 6.2.3 true. If $f(t)$ is finite energy $(f(t) \in L^2[0,T])$, then the equality in Equation 6.2.3 holds in the sense of energy convergence; if $f(t)$ is continuous, then Equation 6.2.3 holds pointwise. Also, if $f(t)$ meets some mild conditions (the Dirichlet conditions), then Equation 6.2.3 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 $f(t)$. The formula shows $f(t)$ as a sum of complex exponentials, each of which is easily processed by an LTI system (since it is an eigenfunction of every LTI system). Mathematically, it tells us that the set of complex exponentials $\{\forall n, n \in \mathbb{Z} : (e^{j\omega_0 n t})\}$ form a basis for the space of T-periodic continuous time functions.

Example 6.2.1

We know from Euler's formula that $\cos(\omega t) + \sin(\omega t) = \frac{1-j}{2} e^{j\omega t} + \frac{1+j}{2} e^{-j\omega t}$.

Synthesis with Sinusoids Demonstration

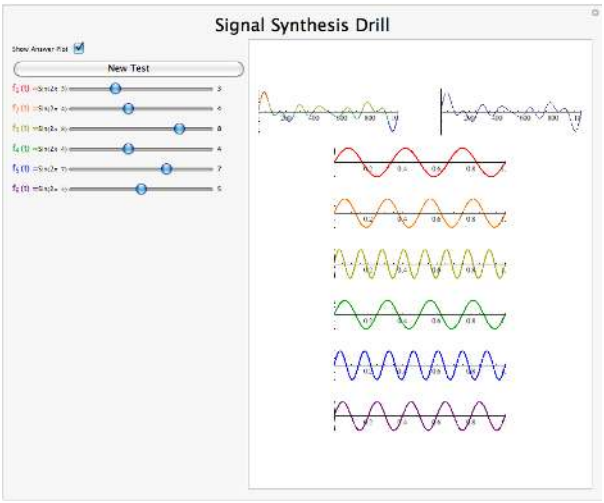


Figure 6.2.2: Interact(when online) with a Mathematica CDF demonstrating sinusoid synthesis. To download, right click and save as .cdf.

Fourier Series Analysis

Finding the coefficients of the Fourier series expansion involves some algebraic manipulation of the synthesis formula. First of all we will multiply both sides of the equation by $e^{-(jω_0kt)}$, where $k \in \mathbb{Z}$.

$$f(t)e^{-(j\omega_0 kt)} = \sum_{n=-\infty}^{\infty} c_n e^{j\omega_0 nt} e^{-(j\omega_0 kt)} \tag{6.2.4}$$

Now integrate both sides over a given period, T:

$$\int_0^T f(t)e^{-(j\omega_0 kt)}\mathrm{d}t = \int_0^T \sum_{n=-\infty}^{\infty} c_n e^{j\omega_0 nt} e^{-(j\omega_0 kt)}\mathrm{d}t \tag{6.2.5}$$

On the right-hand side we can switch the summation and integral and factor the constant out of the integral.

$$\int_0^T f(t)e^{-(j\omega_0 kt)}\mathrm{d}t = \sum_{n=-\infty}^{\infty} c_n \int_0^T e^{j\omega_0 (n-k)t}\mathrm{d}t \tag{6.2.6}$$

Now that we have made this seemingly more complicated, let us focus on just the integral, $\int_0^T e^{j\omega_0(n-k)t}\mathrm{d}t$, 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:

$$\forall n, n = k : \left(\int_0^T e^{j\omega_0(n-k)t}\mathrm{d}t = T\right) \tag{6.2.7}$$

For $n \neq k$, we will have:

$$\forall n, n \neq k : \left(\int_0^T e^{j\omega_0(n-k)t}\mathrm{d}t = \int_0^T \cos\left(\omega_0(n-k)t\right)\mathrm{d}t + j\int_0^T \sin\left(\omega_0(n-k)t\right)\mathrm{d}t\right) \tag{6.2.8}$$

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

$$\int_0^T \cos\left(\omega_0(n-k)t\right)\mathrm{d}t = 0$$

which also holds for the integral involving the sine function. Therefore, we conclude the following about our integral of interest:

$$\int_0^T e^{j\omega_0(n-k)t}\mathrm{d}t = \begin{cases} T \text{ if } n = k \\ 0 \text{ otherwise} \end{cases}$$

Now let us return our attention to our complicated equation, Equation 6.2.6, 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 Equation 6.2.6 will have a nonzero result is when k and n are equal:

$$\forall n, n = k : \left(\int_0^T f(t)e^{-(j\omega_0 nt)}\mathrm{d}t = Tc_n\right)$$

Finally, we have our general equation for the Fourier coefficients:

$$c_n = \frac{1}{T}\int_0^T f(t)e^{-(j\omega_0 nt)}\mathrm{d}t$$

Example 6.2.2

Consider the square wave function given by

$$x(t) = \begin{cases} 1/2 & t \leq 1/2 \\ -1/2 & t > 1/2 \end{cases}$$

on the unit interval $t \in \mathbb{Z}[0, 1)$.

$$
\begin{aligned}
c_k &= \int_0^1 x(t)e^{-j2\pi kt}dt \\
&= \int_0^{1/2} \frac{1}{2}e^{-j2\pi kt}dt - \int_{1/2}^1 \frac{1}{2}e^{-j2\pi kt}dt \\
&= \frac{j\left(-1+e^{j\pi k}\right)}{2\pi k}
\end{aligned}
$$

Thus, the Fourier coefficients of this function found using the Fourier series analysis formula are

$$
c_k = \begin{cases} -j/\pi k & \text{k odd} \\ 0 & \text{k even} \end{cases}
$$

Fourier Series Summary

Because complex exponentials are eigenfunctions of LTI systems, it is often useful to represent signals using a set of complex exponentials as a basis. The continuous time Fourier series synthesis formula expresses a continuous time, periodic function as the sum of continuous time, discrete frequency complex exponentials.

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

The continuous time Fourier series analysis formula gives the coefficients of the Fourier series expansion.

$$
c_n = \frac{1}{T}\int_0^T f(t)e^{-(j\omega_0 nt)}\mathrm{d}t
$$

In both of these equations $\omega_0 = \frac{2\pi}{T}$ is the fundamental frequency.

This page titled 6.2: Continuous Time Fourier Series (CTFS) is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

6.3: Common Fourier Series

Introduction

Once one has obtained a solid understanding of the fundamentals of **Fourier series analysis** and the **General Derivation of the Fourier Coefficients**, it is useful to have an understanding of the common signals used in Fourier Series Signal Approximation.

Deriving the Fourier Coefficients

Consider a square wave $f(x)$ of length 1. Over the range [0,1), this can be written as

$$x(t) = \begin{cases} 1 & t \leq \frac{1}{2} \\ -1 & t > \frac{1}{2} \end{cases} .$$

Fourier series approximation of a square wave

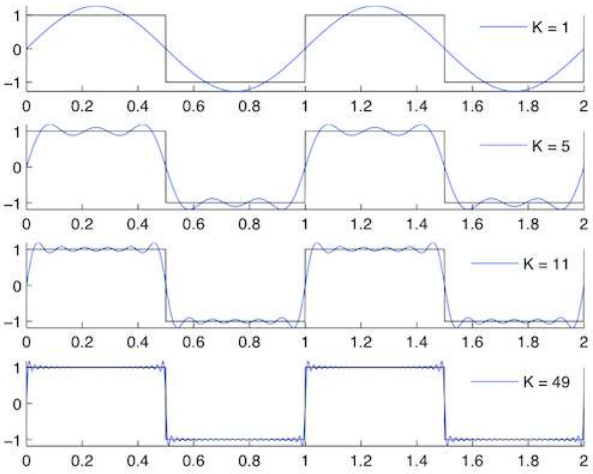


Figure 6.3.1: Fourier series approximation to $sq(t)$. The number of terms in the Fourier sum is indicated in each plot, and the square wave is shown as a dashed line over two periods.

Real Even Signals

Given that the square wave is a real and even signal,

- $f(t) = f(-t)$ EVEN
- $f(t) = f^*(t)$ REAL

therefore,

- $c_n = c_{-n}$ EVEN
- $c_n = c_n^*$ REAL

Consider this mathematical question intuitively: Can a discontinuous function, like the square wave, be expressed as a sum, even an infinite one, of continuous signals? One should at least be suspicious, and in fact, it can't be thus expressed.

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.

Deriving the Fourier Coefficients for Other Signals

The Square wave is the standard example, but other important signals are also useful to analyze, and these are included here.

Constant Waveform

This signal is relatively self-explanatory: the time-varying portion of the Fourier Coefficient is taken out, and we are left simply with a constant function over all time.

$$x(t) = 1$$

Sinusoid Waveform

With this signal, only a specific frequency of time-varying Coefficient is chosen (given that the Fourier Series equation includes a sine wave, this is intuitive), and all others are filtered out, and this single time-varying coefficient will exactly match the desired signal.

$$x(t) = \sin(\pi t)$$

Triangle Waveform

$$x(t) = \begin{cases} t & t \leq 1/4 \\ 2 - 4t & 1/4 \leq t \leq 3/4 \\ -7/4 + 4t & 3/4 \leq t \leq 1 \end{cases}$$

This is a more complex form of signal approximation to the square wave. Because of the **Symmetry Properties** of the Fourier Series, the triangle wave is a real and odd signal, as opposed to the real and even square wave signal. This means that

- $f(t) = -f(-t)$ ODD
- $f(t) = f^*(t)$ REAL

therefore,

- $c_n = -c_{-n}$
- $c_n = -c_n^*$ IMAGINARY

Fourier series approximation of a triangle wave

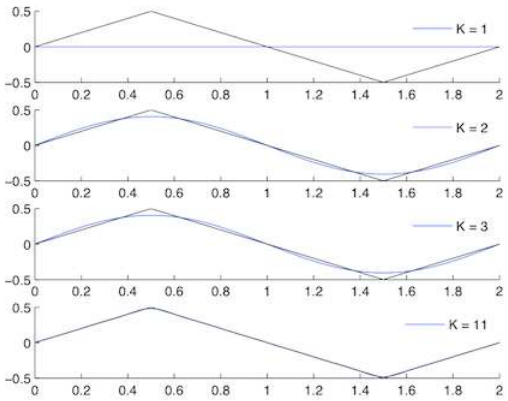


Figure
6.3.2

Sawtooth Waveform

$$x(t) = t - \text{Floor}(t)$$

Because of the **Symmetry Properties** of the Fourier Series, the sawtooth wave can be defined as a real and odd signal, as opposed to the real and even square wave signal. This has important implications for the Fourier Coefficients.

Fourier series approximation of a sawtooth wave

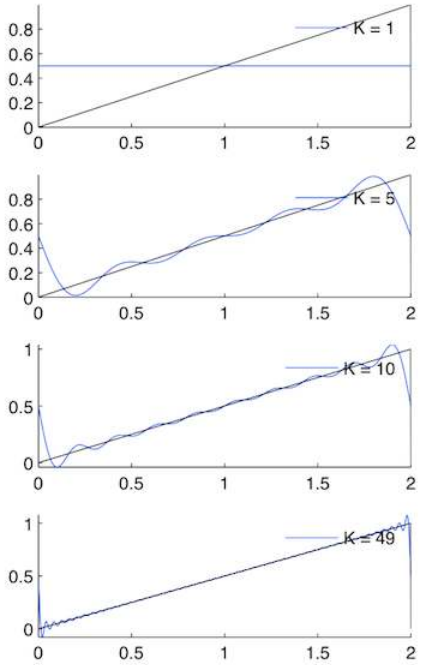


Figure
6.3.3

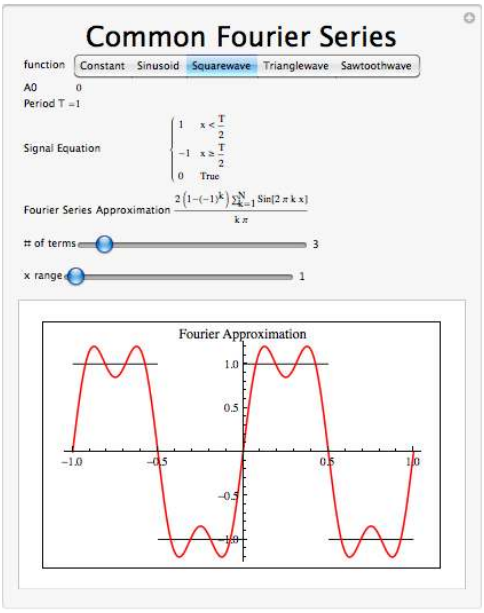


Figure 6.3.4: Interact (when online) with a Mathematica CDF demonstrating the common Fourier Series. To download, right click and save file as .cdf.

Summary

To summarize, a great deal of variety exists among the common Fourier Transforms. A summary table is provided here with the essential information.

Table 6.3.1: Common Continuous-Time Fourier Series

Description	Time Domain Signal for $t \in [0,1)$	Frequency Domain Signal
Constant Waveform	$x(t) = 1$	$c_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$
Sinusoid Waveform	$x(t) = \sin(\pi t)$	$c_k = \begin{cases} 1/2 & k = \pm 1 \\ 0 & k \neq \pm 1 \end{cases}$
Square Waveform	$x(t) = \begin{cases} 1 & t \leq 1/2 \\ -1 & t > 1/2 \end{cases}$	$c_k = \begin{cases} 4/\pi k & \text{k odd} \\ 0 & \text{k even} \end{cases}$
Triangle Waveform	$x(t) = \begin{cases} t & t \leq 1/2 \\ 1-t & t > 1/2 \end{cases}$	$c_k = \begin{cases} -8\sin(\text{k}\pi)/2)/(\pi k)^2 & \text{k odd} \\ 0 & \text{k even} \end{cases}$
Sawtooth Waveform	$x(t) = t/2$	$c_k = \begin{cases} 0.5 & k = 0 \\ -1/\pi k & k \neq 0 \end{cases}$

This page titled 6.3: Common Fourier Series is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

6.4: Properties of the CTFS

Introduction

In this module we will discuss the basic properties of the Continuous-Time Fourier Series. We will begin by refreshing your memory of our basic Fourier series equations:

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

$$c_n = \frac{1}{T}\int_0^T f(t)e^{-(j\omega_0 nt)}\mathrm{d}t$$

Let $\mathscr{F}(\cdot)$ denote the transformation from $f(t)$ to the Fourier coefficients

$$\mathscr{F}(f(t)) = \forall n, n \in \mathbb{Z} : (c_n)$$

$\mathscr{F}(\cdot)$ maps complex valued functions to sequences of complex numbers.

Linearity

$\mathscr{F}(\cdot)$ is a **linear transformation**.

Theorem 6.4.1

If $\mathscr{F}(f(t)) = c_n$ and $\mathscr{F}(g(t)) = d_n$. Then

$$\forall \alpha, \alpha \in \mathbb{C} : (\mathscr{F}(\alpha f(t)) = \alpha c_n)$$

and

$$\mathscr{F}(f(t) + g(t)) = c_n + d_n$$

Proof

Easy. Just linearity of integral.

$$\begin{aligned}\mathscr{F}(f(t)+g(t)) &= \forall n, n \in \mathbb{Z} : \left(\frac{1}{T}\int_0^T (f(t)+g(t))e^{-(j\omega_0 nt)}\mathrm{d}t\right)\\ &= \forall n, n \in \mathbb{Z} : \left(\frac{1}{T}\int_0^T f(t)e^{-(j\omega_0 nt)}\mathrm{d}t + \frac{1}{T}\int_0^T g(t)e^{-(j\omega_0 nt)}\mathrm{d}t\right)\\ &= \forall n, n \in \mathbb{Z} : (c_n + d_n)\\ &= c_n + d_n\end{aligned} \tag{6.4.1}$$

Shifting

Shifting in time equals a phase shift of Fourier coefficients.

Theorem 6.4.2

$\mathcal{F}\left(f\left(t-t_0\right)\right) = e^{-(j\omega_0 nt_0)}c_n$ if $c_n = |c_n|\, e^{j\angle(c_n)}$, then

$$\left|e^{-(j\omega_0 nt_0)}c_n\right| = \left|e^{-(j\omega_0 nt_0)}\right||c_n| = |c_n|$$

$$\angle\left(e^{-(i\omega_0 t_0 n)}\right) = \angle(c_n) - \omega_0 t_0 n$$

Proof

$$\begin{aligned}\mathscr{F}\left(f\left(t-t_0\right)\right) &= \forall n, n \in \mathbb{Z}: \left(\frac{1}{T}\int_0^T f\left(t-t_0\right) e^{-\left(j\omega_0 nt\right)} \mathrm{d}t\right) \\ &= \forall n, n \in \mathbb{Z}: \left(\frac{1}{T}\int_{-t_0}^{T-t_0} f\left(t-t_0\right) e^{-\left(j\omega_0 n(t-t_0)\right)} e^{-\left(j\omega_0 nt_0\right)} \mathrm{d}t\right) \\ &= \forall n, n \in \mathbb{Z}: \left(\frac{1}{T}\int_{-t_0}^{T-t_0} f(\tilde{t}) e^{-\left(j\omega_0 n\tilde{t}\right)} e^{-\left(j\omega_0 nt_0\right)} \mathrm{d}t\right) \\ &= \forall n, n \in \mathbb{Z}: \left(e^{-\left(j\omega_0 n\tilde{t}\right)} c_n\right) \end{aligned} \tag{6.4.2}$$

Parseval's Relation

$$\int_0^T \left(|f(t)|\right)^2 \mathrm{d}t = T\sum_{n=-\infty}^{\infty} \left(|c_n|\right)^2$$

Parseval's relation tells us that the energy of a signal is equal to the energy of its Fourier transform.

Note

Parseval tells us that the Fourier series **maps** $L^2([0,T])$ to $l^2(\mathbb{Z})$.

$$f(t) \xleftrightarrow{F.S.} \{c_n\}$$

$$f(t) \in L^2[0,T] \Longleftrightarrow \{c_n\} \in l^2\{\mathbb{Z}\}$$

Figure 6.4.1

Exercise 6.4.1

For $f(t)$ to have "finite energy," what do the c_n do as $n \to \infty$?

Answer

$\left(|c_n|\right)^2 < \infty$ for $f(t)$ to have finite energy.

Exercise 6.4.2

If $\forall n, |n| > 0 : \left(c_n = \frac{1}{n}\right)$, is $f \in L^2([0,T])$?

Answer

Yes, because $\left(|c_n|\right)^2 = \frac{1}{n}$, which is summable.

Exercise 6.4.3

Now, if $\forall n, |n| > 0 : \left(c_n = \frac{1}{\sqrt{n}}\right)$ is $f \in L^2([0,T])$?

Answer

No, because $\left(|c_n|\right)^2 = \frac{1}{n}$, which is not summable.

The rate of decay of the Fourier series determines if $f(t)$ has **finite energy**.

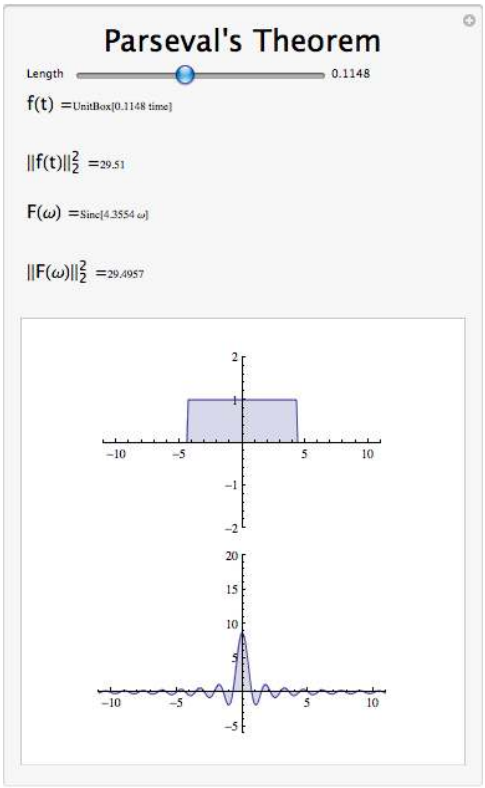


Figure 6.4.2: Interact (when online) with a Mathematica CDF demonstrating Parsevals Theorem. To download, right click and save file as .cdf.

Symmetry Properties

Rule 6.4.1: Even Signals

Even Signals

$f(t) = f(-t)$
$\|c_n\| = \|c_{-n}\|$

Proof

$c_n = \frac{1}{T}\int_0^T f(t)\exp(-j\omega_0 nt)\,dt$
$= \frac{1}{T}\int_0^{\frac{T}{2}} f(t)\exp(-j\omega_0 nt)\,dt + \frac{1}{T}\int_{\frac{T}{2}}^T f(t)\exp(-j\omega_0 nt)\,dt$
$= \frac{1}{T}\int_0^{\frac{T}{2}} f(-t)\exp(-j\omega_0 nt)\,dt + \frac{1}{T}\int_{\frac{T}{2}}^T f(-t)\exp(-j\omega_0 nt)\,dt$
$= \frac{1}{T}\int_0^T f(t)\left[\exp(j\omega_0 nt)\,dt + \exp(-j\omega_0 nt)\right]dt$
$= \frac{1}{T}\int_0^T f(t)2\cos(\omega_0 nt)\,dt$

Rule 6.4.2: Odd Signals

Odd Signals

$f(t) = -f(-t)$
$c_n = c_{-n}^*$

Proof

$c_n = \frac{1}{T}\int_0^T f(t)\exp(-j\omega_0 nt)\,dt$
$= \frac{1}{T}\int_0^{\frac{\pi}{2}} f(t)\exp(-j\omega_0 nt)\,dt + \frac{1}{T}\int_{\frac{T}{2}}^T f(t)\exp(-j\omega_0 nt)\,dt$
$= \frac{1}{T}\int_0^{\frac{T}{2}} f(t)\exp(-j\omega_0 nt)\,dt - \frac{1}{T}\int_{\frac{T}{2}}^T f(-t)\exp(j\omega_0 nt)\,dt$
$= -\frac{1}{T}\int_0^T f(t)\left[\exp(j\omega_0 nt)\,dt - \exp(-j\omega_0 nt)\right]dt$
$= -\frac{1}{T}\int_0^T f(t)2j\sin(\omega_0 nt)\,dt$

Rule 6.4.3: Real Signals

Real Signals

$f(t) = f^*(t)$
$c_n = c_{-n}^*$

Proof

$c_n = \frac{1}{T}\int_0^T f(t)\exp(-j\omega_0 nt)\,dt$
$= \frac{1}{T}\int_0^{\frac{T}{2}} f(t)\exp(-j\omega_0 nt)\,dt + \frac{1}{T}\int_{\frac{T}{2}}^T f(t)\exp(-j\omega_0 nt)\,dt$
$= \frac{1}{T}\int_0^{\frac{T}{2}} f(-t)\exp(-j\omega_0 nt)\,dt + \frac{1}{T}\int_{\frac{T}{2}}^T f(-t)\exp(-j\omega_0 nt)\,dt$
$= \frac{1}{T}\int_0^T f(t)\left[\exp(j\omega_0 nt)\,dt + \exp(-j\omega_0 nt)\right]dt$
$= \frac{1}{T}\int_0^T f(t)2\cos(\omega_0 nt)\,dt$

Differentiation in Fourier Domain

$$(\mathcal{F}(f(t)) = c_n) \Rightarrow \left(\mathcal{F}\left(\frac{df(t)}{dt}\right) = jn\omega_0 c_n\right)$$

Since

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

then

https://eng.libretexts.org/@go/page/22875

$$\frac{d}{dt}f(t) = \sum_{n=-\infty}^{\infty} c_n \frac{de^{j\omega_0 nt}}{dt}$$
$$= \sum_{n=-\infty}^{\infty} c_n j\omega_0 n e^{i\omega_0 nt} \tag{6.4.3}$$

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 $n \to \infty$. If $\mathscr{F}(f(t)) = c_n$ and $|c_n|$ has the form $\frac{1}{n^k}$, then $\mathscr{F}\left(\frac{\mathrm{d}^m f(t)}{\mathrm{d}t^m}\right) = (jn\omega_0)^m c_n$ and has the form $\frac{n^m}{n^k}$. So for the m^{th} derivative to have finite energy, we need

$$\sum_n \left(\left|\frac{n^m}{n^k}\right|\right)^2 < \infty$$

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.

Fourier Differentiation Demonstration

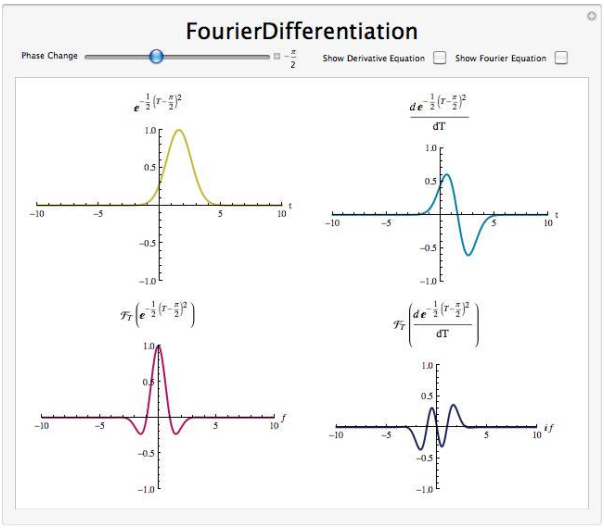


Figure 6.4.3: Interact (when online) with a Mathematica CDF demonstrating Differentiation in the Fourier Domain. To download, right click and save file as .cdf.

Integration in the Fourier Domain

If

$$\mathscr{F}(f(t)) = c_n$$

then

$$\mathscr{F}\left(\int_{-\infty}^{t} f(\tau)\mathrm{d}\tau\right) = \frac{1}{j\omega_0 n}c_n$$

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.

Fourier Integration Demonstration

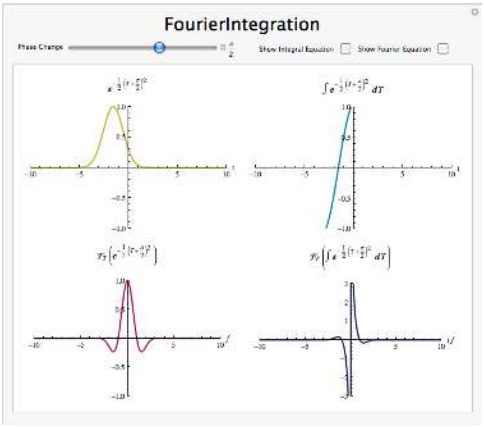


Figure 6.4.4: Interact (when online) with a Mathematica CDF demonstrating Integration in the Fourier Domain. To download, right click and save file as .cdf.

Signal Multiplication and 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, $y(t)$, where $y(t) = f(t)g(t)$. We find that the Fourier Series representation of $y(t)$, e_n, is such that $e_n = \sum_{i=-\infty}^{\infty} c_k d_{n-k}$. This is to say that signal multiplication in the time domain is equivalent to **signal convolution** in the frequency domain, and vice-versa: signal multiplication in the frequency domain is equivalent to signal convolution in the time domain. The proof of this is as follows

$$\begin{aligned} e_n &= \frac{1}{T}\int_0^T f(t)g(t)e^{-(j\omega_0 nt)}\mathrm{d}t \\ &= \frac{1}{T}\int_0^T \sum_{k=-\infty}^{\infty} c_k e^{j\omega_0 kt} g(t) e^{-(j\omega_0 nt)}\mathrm{d}t \\ &= \sum_{k=-\infty}^{\infty} c_k \left(\frac{1}{T}\int_0^T g(t)e^{-(j\omega_0(n-k)t)}\mathrm{d}t\right) \\ &= \sum_{k=-\infty}^{\infty} c_k d_{n-k} \end{aligned} \tag{6.4.4}$$

for more details, see the section on Signal convolution and the CTFS (Section 4.3).

Conclusion

Like other Fourier transforms, the CTFS has many useful properties, including linearity, equal energy in the time and frequency domains, and analogs for shifting, differentiation, and integration.

Table 6.4.1: Properties of the CTFS

Property	Signal	CTFS
Linearity	$ax(t) + by(t)$	$aX(f) + bY(f)$
Time Shifting	$x(t-\tau)$	$X(f)e^{-j2\pi f\tau/T}$
Time Modulation	$x(t)e^{j2\pi f\tau/T}$	$X(f-k)$
Multiplication	$x(t)y(t)$	$X(f) * Y(f)$
Continuous Convolution	$x(t) * y(t)$	$X(f)Y(f)$

This page titled 6.4: Properties of the CTFS is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

6.5: Continuous Time Circular Convolution and the CTFS

Introduction

This module relates circular convolution of periodic signals in the time domain to multiplication in the frequency domain.

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 $v(t) = f(t) \circledast g(t)$. We find that the Fourier Series representation of $v(t)$, a_n, is such that $a_n = c_n d_n$. $f(t) \circledast g(t)$ is the circular convolution (Section 7.5) of two periodic signals and is equivalent to the convolution over one interval, i.e. $f(t) \circledast g(t) = \int_0^T \int_0^T f(\tau) g(t-\tau) d\tau dt$.

Note

Circular convolution in the time domain is equivalent to multiplication of the Fourier coefficients.

This is proved as follows

$$\begin{aligned}
a_n &= \frac{1}{T}\int_0^T v(t)e^{-(j\omega_0 nt)}\mathrm{d}t \\
&= \frac{1}{T^2}\int_0^T \int_0^T f(\tau)g(t-\tau)\mathrm{d}\tau e^{-(\omega j_0 nt)}\mathrm{d}t \\
&= \frac{1}{T}\int_0^T f(\tau)\left(\frac{1}{T}\int_0^T g(t-\tau)e^{-(j\omega_0 nt)}\mathrm{d}t\right)\mathrm{d}\tau \\
&= \forall \nu, \nu = t-\tau : \left(\frac{1}{T}\int_0^T f(\tau)\left(\frac{1}{T}\int_{-\tau}^{T-\tau} g(\nu)e^{-(j\omega_0(\nu+\tau))}\mathrm{d}\nu\right)\mathrm{d}\tau\right) \\
&= \frac{1}{T}\int_0^T f(\tau)\left(\frac{1}{T}\int_{-\tau}^{T-\tau} g(\nu)e^{-(j\omega_0 n\nu)}\mathrm{d}\nu\right)e^{-(j\omega_0 n\tau)}\mathrm{d}\tau \\
&= \frac{1}{T}\int_0^T f(\tau)d_n e^{-(j\omega_0 n\tau)}\mathrm{d}\tau \\
&= d_n\left(\frac{1}{T}\int_0^T f(\tau)e^{-(j\omega_0 n\tau)}\mathrm{d}\tau\right) \\
&= c_n d_n
\end{aligned} \tag{6.5.1}$$

Exercise

Take a look at a square pulse with a period of T.

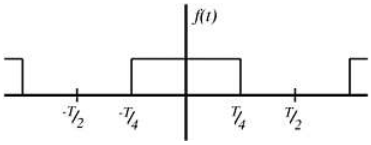


Figure 6.5.1

For this signal

$$c_n = \begin{cases} \frac{1}{T} \text{ if } n = 0 \\ \frac{1}{2}\frac{\sin(\frac{\pi}{2}n)}{\frac{\pi}{2}n} \text{ otherwise} \end{cases}$$

Take a look at a triangle pulse train with a period of T.

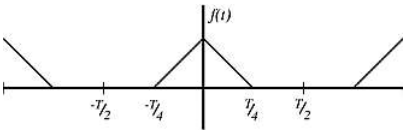


Figure 6.5.2

This signal is created by circularly convolving the square pulse with itself. The Fourier coefficients for this signal are $a_n = c_n^2 = \frac{1}{4}\frac{\sin^2}{(\frac{\pi}{2}n)}$.

Exercise 6.5.1

Find the Fourier coefficients of the signal that is created when the square pulse and the triangle pulse are convolved.

Answer

$$a_n = \begin{cases} \text{undefined} & n = 0 \\ \frac{1}{8}\frac{\sin^3(\frac{\pi}{2}n)}{(\frac{\pi}{2}n)^3} & \text{otherwise} \end{cases}$$

Conclusion

Circular convolution in the time domain is equivalent to multiplication of the Fourier coefficients in the frequency domain.

This page titled 6.5: Continuous Time Circular Convolution and the CTFS is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

6.6: Convergence of Fourier Series

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 14.4) or eigenfunctions of LTI systems (Section 14.5). We have shown that we can represent a signal as the sum of exponentials through the Fourier Series equations below:

$$f(t) = \sum_n c_n e^{j\omega_0 nt} \tag{6.6.1}$$

$$c_n = \frac{1}{T}\int_0^T f(t)e^{-(j\omega_0 nt)}dt \tag{6.6.2}$$

Joseph Fourier 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 Equation 6.6.1 (indeed he proved that Equation 6.6.1 holds for continuous-time functions). However, we know now that the real truth lies in between Fourier and Lagrange's positions.

Understanding the Truth

Formulating our question mathematically, let

$$f'_N(t) = \sum_{n=-N}^{N} c_n e^{j\omega_0 nt}$$

where c_n equals the Fourier coefficients of $f(t)$ (see Equation 6.6.2).

$f'_N(t)$ is a "partial reconstruction" of $f(t)$ using the first $2N+1$ Fourier coefficients. f'_N **approximates** $f(t)$, with the approximation getting better and better as N gets large. Therefore, we can think of the set $\left\{\forall N, N = \{0,1,...\} : \left(\frac{\mathrm{d}f_N(t)}{\mathrm{d}}\right)\right\}$ 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 $f'_N(t) \to f(t)$ as $N \to \infty$? 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 $\lim_{N\to\infty} \frac{df_N(t)}{d}$ at **each point** and comparing to $f(t)$.

Approach #1

Let $e_N(t)$ be the difference (i.e. error) between the signal $f(t)$ and its partial reconstruction f'_N

$$e_N(t) = f(t) - f'_N(t)$$

If $f(t) \in L^2([0,T])$ (finite energy), then the energy of $e_N(t) \to 0$ as $N \to \infty$ is

$$\int_0^T \left(|e_N(t)|\right)^2 \mathrm{d}t = \int_0^T \left(f(t) - f'_N(t)\right)^2 \mathrm{d}t \to 0$$

We can prove this equation using Parseval's relation:

$$\lim_{N\to\infty}\int_0^T \left(|f(t) - f'_N(t)|\right)^2 \mathrm{d}t = \lim_{N\to\infty}\sum_{N=-\infty}^{\infty}\left(\left|\mathscr{F}_n(f(t)) - \mathscr{F}_n\left(\frac{\mathrm{d}f_N(t)}{\mathrm{d}}\right)\right|\right)^2 = \lim_{N\to\infty}\sum_{|n|>N}\left(|c_n|\right)^2 = 0$$

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.

Approach #2

The fact that $e_N \to 0$ says nothing about $f(t)$ and $\lim_{N\to\infty} \frac{\mathrm{d}f_N(t)}{\mathrm{d}}$ being **equal** at a given point. Take the two functions graphed below for example:

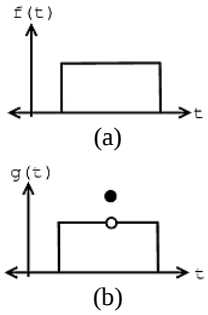


Figure 6.6.1

Given these two functions, $f(t)$ and $g(t)$, then we can see that for all t, $f(t) \neq g(t)$, but

$$\int_0^T (|f(t) - g(t)|)^2 \mathrm{d}t = 0$$

From this we can see the following relationships:

$$\text{energy convergence} \neq \text{pointwise convergence}$$

$$\text{pointwise convergence} \Rightarrow \text{convergence in } L^2([0,T])$$

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

$$f(t_0) \neq \lim_{N\to\infty} \frac{\mathrm{d}f_N(t_0)}{\mathrm{d}}$$

But as long as $f(t)$ meets some other fairly mild conditions, then

$$f(t') = \lim_{N\to\infty} \frac{\mathrm{d}f_N(t'))}{\mathrm{d}}$$

if $f(t)$ is **continuous** at $t = t'$.

These conditions are known as the **Dirichlet Conditions**.

Dirichlet Conditions

Named after the German mathematician, Peter Dirichlet, the **Dirichlet conditions** are the sufficient conditions to guarantee **existence** and **energy convergence** of the Fourier Series.

The Weak Dirichlet Condition for the Fourier Series

For the Fourier Series to exist, the Fourier coefficients must be finite. The **Weak Dirichlet Condition** guarantees this. It essentially says that the integral of the absolute value of the signal must be finite.

Theorem 6.6.1: Weak Dirichlet Condition for the Fourier Series

The coefficients of the Fourier Series are finite if

$$\int_0^T |f(t)|dt < \infty$$

Proof

This can be shown from the magnitude of the Fourier Series coefficients:

$$|c_n| = \left| \frac{1}{T} \int_0^T f(t) e^{-(j\omega_0 nt)} dt \right| \leq \frac{1}{T} \int_0^T \left| f(t) \| e^{-(j\omega_0 nt)} \right| dt$$

Remembering our complex exponentials (Section 1.8), we know that in the above equation $\left|e^{-(j\omega_0 nt)}\right| = 1$, which gives us:

$$|c_n| \leq \frac{1}{T} \int_0^T |f(t)| dt < \infty \tag{6.6.3}$$

$$\Rightarrow (|c_n| < \infty) \tag{6.6.4}$$

Note

If we have the function:

$$\forall t, 0 < t \leq T : \left(f(t) = \frac{1}{t} \right)$$

then you should note that this function **fails** the above condition because:

$$\int_0^T \left| \frac{1}{t} \right| \mathrm{d}t = \infty$$

The Strong Dirichlet Conditions for the Fourier Series

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.

Example 6.6.1

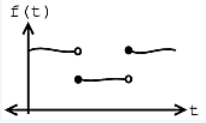
Let us assume we have the following function and equality:

$$f'(t) = \lim_{N \to \infty} \frac{\mathrm{d} f_N(t)}{\mathrm{d}}$$

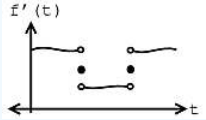
If $f(t)$ meets all three conditions of the Strong Dirichlet Conditions, then

$$f(\tau) = f'(\tau)$$

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.



(a)



(b)

Figure 6.6.2: Discontinuous functions, $f(t)$

Note

The functions that fail the strong Dirchlet conditions are pretty pathological - as engineers, we are not too interested in them.

This page titled 6.6: Convergence of Fourier Series is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

6.7: Gibbs Phenomena

Introduction

The Fourier Series is the representation of continuous-time, periodic signals in terms of complex exponentials. The Dirichlet conditions 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 1.8) 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.

History

In the late 1800s, many machines were built to calculate Fourier coefficients and re-synthesize:

$$f'_N(t) = \sum_{n=-N}^{N} c_n e^{j\omega_0 nt}$$

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

$$f'_{79}(t) = \sum_{n=-79}^{79} c_n e^{j\omega_0 nt}$$

The machine performed very well on all tests except those involving **discontinuous functions**. When a square wave, like that shown in Figure 6.7.1, 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 6.7.1. J. Willard Gibbs first explained this phenomenon in 1899, and therefore these discontinuous points are referred to as **Gibbs Phenomenon**.

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.

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.

$$s(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos\left(\frac{2\pi kt}{T}\right) + \sum_{k=1}^{\infty} b_k \sin\left(\frac{2\pi kt}{T}\right)$$

Figure 6.7.1 shows several Fourier series approximations of the square wave using a varied number of terms, denoted by KK:

Fourier series approximations of a square wave

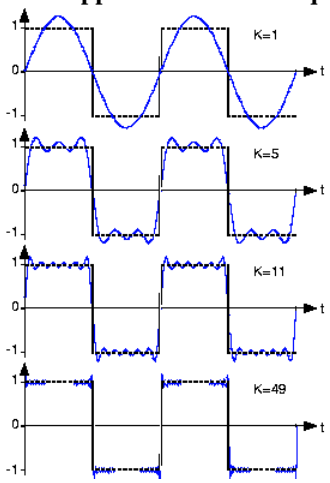


Figure 6.7.1: Fourier series approximation to $sq(t)$. The number of terms in the Fourier sum is indicated in each plot, and the square wave is shown as a dashed line over two periods.

When comparing the square wave to its Fourier series representation in Figure 6.7.1, 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 6.7.1 does reveal a potential issue: Does the Fourier series really equal the square wave at all values of tt? 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 Fourier 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 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.

Redefine Equality

Let's return to the question of equality; how can the equal sign in the definition of the Fourier series (Section 4.3) 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.

$$\lim_{K\to\infty} \text{rms}\left(\varepsilon_K\right) = 0$$

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.

Visualizing Gibb's Phenomena

The following VI demonstrates the occurrence of Gibb's Phenomena. Note how the wiggles near the square pulse to the left remain even if you drastically increase the order of the approximation, even though they do become narrower. Also notice how the

approximation of the smooth region in the middle is much better than that of the discontinuous region, especially at lower orders.

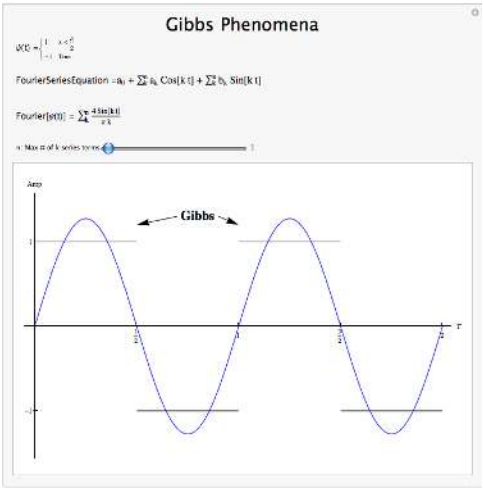


Figure 6.7.2: Interact (when online) with a Mathematica CDF demonstrating Gibbs Phenomena. To download, right click and save as .cdf.

Conclusion

We can **approximate** a function by re-synthesizing using only some of the Fourier coefficients (truncating the F.S.)

$$f'_N(t) = \sum_{nn \leq |N|} c_n e^{j\omega_0 nt}$$

This approximation works well where $f(t)$ is continuous, but not so well where $f(t)$ is discontinuous. In the regions of discontinuity, we will always find Gibb's Phenomena, which never decrease below 9% of the height of the discontinuity, but become narrower and narrower as we add more terms.

This page titled 6.7: Gibbs Phenomena is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

7: Discrete Time Fourier Series (DTFS)

7.1: Discrete Time Periodic Signals

7.2: Discrete Time Fourier Series (DTFS)

7.3: Common Discrete Fourier Series

7.4: Properties of the DTFS

7.5: Discrete Time Circular Convolution and the DTFS

This page titled 7: Discrete Time Fourier Series (DTFS) is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

7.1: Discrete Time Periodic Signals

Introduction

This module describes the type of signals acted on by the Discrete Time Fourier Series.

Relevant Spaces

The Discrete Time Fourier Series maps finite-length (or N-periodic), discrete time signals in L^2 to finite-length, discrete-frequency signals in l^2.

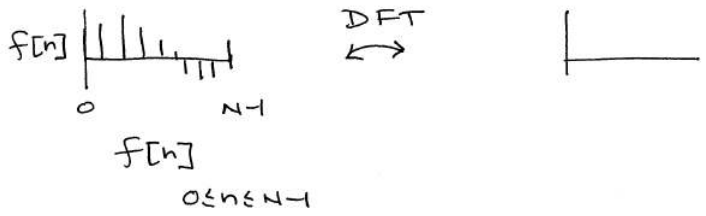


Figure 7.1.1

Periodic signals in discrete time repeats themselves in each cycle. However, only integers are allowed as time variable in discrete time. We denote signals in such case as $x[n]$, $n = \ldots, -2, -1, 0, 1, 2, \ldots$

Periodic Signals

When a function repeats itself exactly after some given period, or cycle, we say it's **periodic**. A **periodic function** can be mathematically defined as:

$$f[n] = f[n + mN] \forall m : (m \in \mathbb{Z}) \tag{7.1.1}$$

where $N > 0$ represents the **fundamental period** of the signal, which is the smallest positive value of N for the signal to repeat. Because of this, you may also see a signal referred to as an N-periodic signal. Any function that satisfies this equation is said to be **periodic** with period N. Here's an example of a **discrete-time periodic signal** with period N:

discrete-time periodic signal

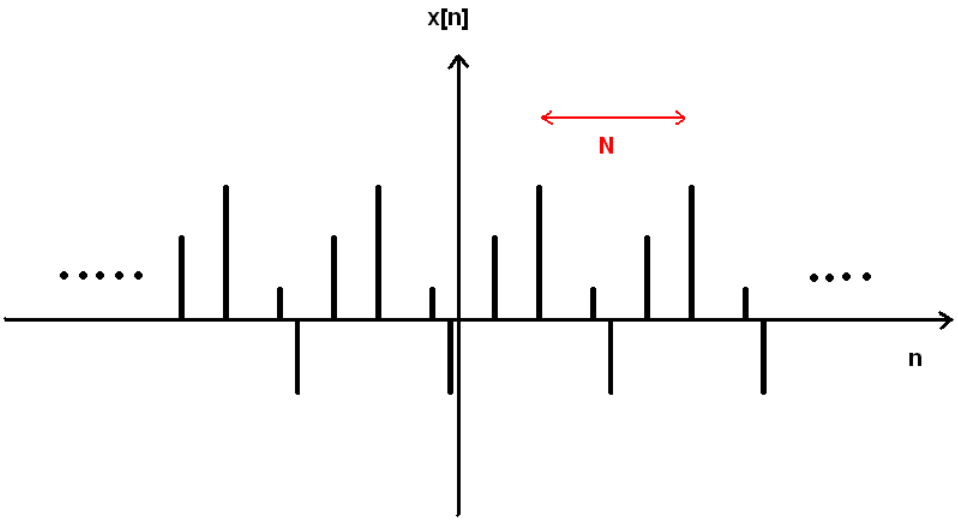


Figure 7.1.2: Notice the function is the same after a time shift of N

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

1. as functions on **all** of $\mathbb{R}$

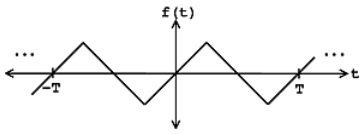


Figure 7.1.3: discrete time periodic function over all of $\mathbb{R}$ where $f[n_0] = f[n_0 + N]$

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

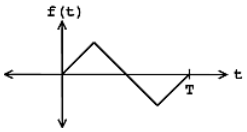


Figure 7.1.4: Remove the redundancy of the period function so that $f[n]$ is undefined outside $[0, N]$.

An **aperiodic** DT function $f[n]$ does not repeat for **any** $N \in \mathbb{R}$; i.e. there exists no N such that Equation 7.1.1 holds.

SinDrillDiscrete Demonstration

Here's an example demonstrating a **periodic** sinusoidal signal with various frequencies, amplitudes and phase delays:

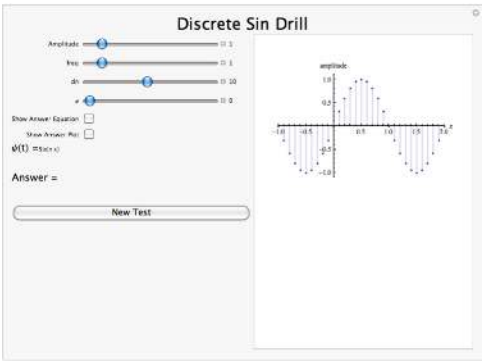


Figure 7.1.5: Interact (when online) with a Mathematica CDF demonstrating a discrete periodic sinusoidal signal with various frequencies, amplitudes and phase delays.

Conclusion

A discrete periodic signal is completely defined by its values in one period, such as the interval [0,N].

This page titled 7.1: Discrete Time Periodic Signals is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

7.2: Discrete Time Fourier Series (DTFS)

Introduction

In this module, we will derive an expansion for discrete-time, periodic functions, and in doing so, derive the **Discrete Time Fourier Series** (DTFS), or the Discrete Fourier Transform (DFT).

DTFS

Eigenfunction analysis

Since complex exponentials (Section 1.8) are eigenfunctions of linear time-invariant (LTI) systems (Section 14.5), calculating the output of an LTI system $\mathscr{H}$ given $e^{j\omega n}$ as an input amounts to simple multiplication, where $\omega_0 = \frac{2\pi k}{N}$, and where $H[k] \in \mathbb{C}$ is the eigenvalue corresponding to k. As shown in the figure, a simple exponential input would yield the output

$$y[n] = H[k]e^{j\omega n}$$

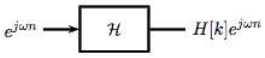


Figure 7.2.1: Simple LTI system.

Using this and the fact that $\mathscr{H}$ is linear, calculating $y[n]$ for combinations of **complex exponentials** is also straightforward.

$$c_1 e^{j\omega_1 n} + c_2 e^{j\omega_2 n} \rightarrow c_1 H\left[k_1\right] e^{j\omega_1 n} + c_2 H\left[k_2\right] e^{j\omega_1 n}$$
$$\sum_l c_l e^{j\omega_1 n} \rightarrow \sum_l c_l H\left[k_l\right] e^{j\omega_l n}$$

The action of H on an input such as those in the two equations above is easy to explain. $\mathscr{H}$ **independently scales** each exponential component $e^{j\omega_l n}$ by a different complex number $H\left[k_l\right] \in \mathbb{C}$. As such, if we can write a function $y[n]$ as a combination of complex exponentials it allows us to easily calculate the output of a system.

DTFS synthesis

It can be demonstrated that an arbitrary Discrete Time-periodic function $f[n]$ can be written as a linear combination of harmonic complex sinusoids

$$f[n] = \sum_{k=0}^{N-1} c_k e^{j\omega_0 k n} \tag{7.2.1}$$

where $\omega_0 = \frac{2\pi}{N}$ is the fundamental frequency. For almost all $f[n]$ of practical interest, there exists c_n to make Equation 7.2.1 true. If $f[n]$ is finite energy ($f[n] \in L^2[0, N]$), then the equality in Equation 7.2.1 holds in the sense of energy convergence; with discrete-time signals, there are no concerns for divergence as there are with continuous-time signals.

The c_n - called the Fourier coefficients - tell us "how much" of the sinusoid $e^{j\omega_0 k n}$ is in $f[n]$. The formula shows $f[n]$ as a sum of complex exponentials, each of which is easily processed by an LTI system (since it is an eigenfunction of every LTI system). Mathematically, it tells us that the set of complex exponentials $\{\forall k, k \in \mathbb{Z} : (e^{j\omega_0 k n})\}$ form a basis for the space of N-periodic discrete time functions.

DFT Synthesis Demonstration

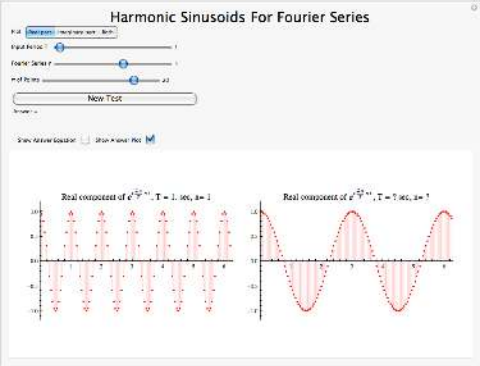


Figure 7.2.2: Download or Interact (when online) with a Mathematica CDF demonstrating Discrete Harmonic Sinusoids. To download, right click and save as .cdf.

DTFS Analysis

Say we have the following set of numbers that describe a periodic, discrete-time signal, where $N = 4$:

$$\{\ldots, 3, 2, -2, 1, 3, \ldots\}$$

Such a periodic, discrete-time signal (with period N) can be thought of as a **finite** set of numbers. For example, we can represent this signal as either a periodic signal or as just a single interval as follows:

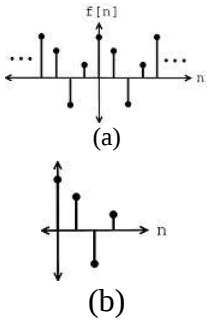


Figure 7.2.3: Here we can look at just one period of the signal that has a vector length of four and is contained in $\mathbb{C}^4$. (a) Periodic Function (b) Function on the interval $[0, T]$

Note

The cardinalsity of the set of discrete time signals with period N equals $\mathbb{C}^N$.

Here, 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.

Complex Sinusoids

If you are familiar with the basic sinusoid signal and with complex exponentials (Section 1.8) then you should not have any problem understanding this section. In most texts, you will see the the discrete-time, complex sinusoid noted as:

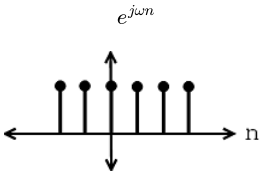


Figure 7.2.4: Complex sinusoid with frequency $\omega = 0$

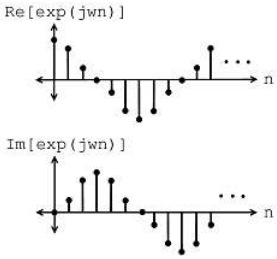


Figure 7.2.5: Complex sinusoid with frequency $\omega = \frac{\pi}{4}$

In the Complex Plane

The complex sinusoid can be directly mapped onto our complex plane, 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:

$$\left|e^{j\omega n}\right| = 1, n \in \mathbb{R}$$

which tells that our complex sinusoid only takes values on the unit circle. As for the angle, the following statement holds true:

$$\angle\left(e^{j\omega n}\right) = wn$$

For more information, see the section on the **Discrete Time Complex Exponential** to learn about **Aliasing**, **Negative Frequencies**, and the formal definition of the **Complex 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 $\forall n, n = \{0, \dots, N-1\} : (b_k = e^{j\omega_k n})$ 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

$e^{j\frac{2\pi}{N}kn}$

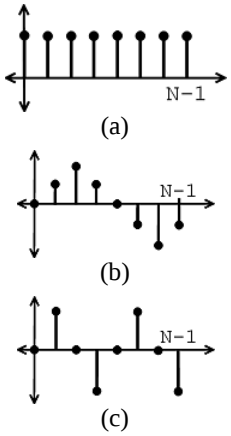


Figure 7.2.6: Examples of our Harmonic Sinusoids (a) Harmonic sinusoid with $k = 0$ (b) Imaginary part of sinusoid, $\text{Im}\left(e^{j\frac{2\pi}{N}1n}\right)$, with $k = 1$ (c) Imaginary part of sinusoid, $\text{Im}\left(e^{j\frac{2\pi}{N}2n}\right)$, with $k = 2$

$e^{j\frac{2\pi}{N}kn}$ is periodic with period N and has k "cycles" between $n = 0$ and $n = N - 1$.

Theorem 7.2.1

If we let

$$b_k[n] = \frac{1}{\sqrt{N}} e^{j\frac{2\pi}{N}kn}, \quad n = \{0, \dots, N-1\}$$

where the exponential term is a vector in $\mathbb{C}^N$, then $\{b_k\}|_{k=\{0,\dots,N-1\}}$ is an orthonormal basis (Section 15.8) for $\mathbb{C}^N$.

Proof

First of all, we must show bk is orthonormal, i.e. $\langle b_k, b_l \rangle = \delta_{kl}$

$$\langle b_k, b_l \rangle = \sum_{n=0}^{N-1} b_k[n] b_l[n]^* = \frac{1}{N} \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}kn} e^{-(j\frac{2\pi}{N}ln)}$$

$$\langle b_k, b_l \rangle = \frac{1}{N} \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(l-k)n}$$

If $l = k$, then

$$\begin{aligned} \langle b_k, b_l \rangle &= \frac{1}{N} \sum_{n=0}^{N-1} 1 \\ &= 1 \end{aligned} \tag{7.2.2}$$

If $l \neq k$ then we must use the "partial summation formula" shown below:

$$\sum_{n=0}^{N-1} \alpha^n = \sum_{n=0}^{\infty} \alpha^n - \sum_{n=N}^{\infty} \alpha^n = \frac{1}{1-\alpha} - \frac{\alpha^N}{1-\alpha} = \frac{1-\alpha^N}{1-\alpha}$$

$$\langle b_k, b_l \rangle = \frac{1}{N} \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(l-k)n}$$

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.

$$\langle b_k, b_l \rangle = \frac{1}{N} \frac{1 - e^{j\frac{2\pi}{N}(l-k)N}}{1 - e^{j\frac{2\pi}{N}(l-k)}} = \frac{1}{N} \frac{1-1}{1 - e^{j\frac{2\pi}{N}(l-k)}} = 0$$

So,

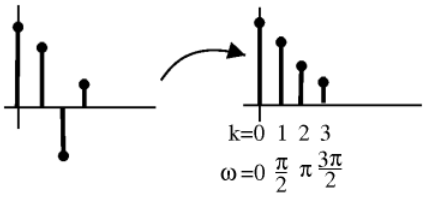
$$\langle b_k, b_l \rangle = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

Therefore: b_k is an orthonormal set. b_k is also a basis (Section 14.1), since there are N vectors which are linearly independent (Section 14.1 - Linear Independence) (orthogonality implies linear independence).

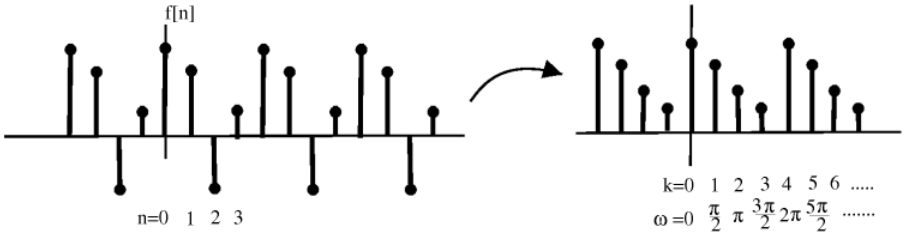
And finally, we have shown that the harmonic sinusoids $\left\{ \frac{1}{\sqrt{N}} e^{j\frac{2\pi}{N}kn} \right\}$ form an orthonormal basis for $\mathbb{C}^n$

Periodic Extension to DTFS

Now that we have an understanding of the discrete-time Fourier series (DTFS), we can consider the **periodic extension** of $c[k]$ (the Discrete-time Fourier coefficients). Figure 7.2.7 shows a simple illustration of how we can represent a sequence as a periodic signal mapped over an infinite number of intervals.



(a)



(b)

Figure 7.2.7: (a) vectors (b) periodic sequences

Exercise 7.2.1

Why does a periodic (Section 6.1) extension to the DTFS coefficients $c[k]$ make sense?

Answer

Aliasing: $b_k = e^{j\frac{2\pi}{N}kn}$

$$\begin{aligned} b_{k+N} &= e^{j\frac{2\pi}{N}(k+N)n} \\ &= e^{j\frac{2\pi}{N}kn} e^{j2\pi n} \\ &= e^{j\frac{2\pi}{N}n} \\ &= b_k \end{aligned} \tag{7.2.3}$$

Example 7.2.1: Discrete Time Square Wave

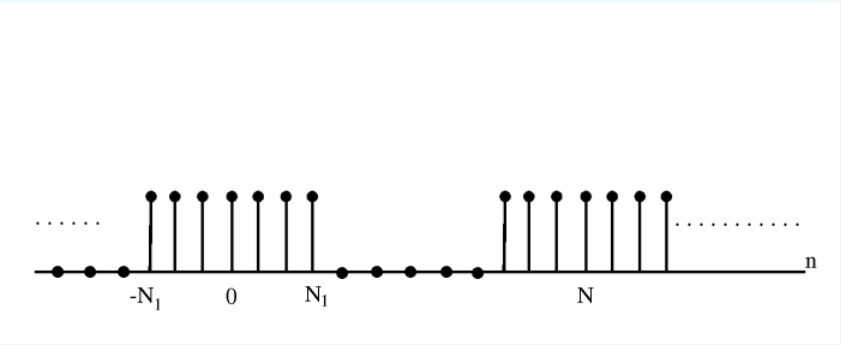


Figure 7.2.8

the DTFS $c[k]$ using:

$$c[k] = \frac{1}{N}\sum_{n=0}^{N-1} f[n]e^{-(j\frac{2\kappa}{N}kn)}$$

Just like continuous time Fourier series, we can take the summation over any interval, so we have

$$c_k = \frac{1}{N}\sum_{n=-N_1}^{N_1} e^{-(j\frac{2\pi}{N}kn)}$$

Let $m = n + N_1$ (so we can get a geometric series starting at 0)

$$\sum_{n=0}^{M} a^n = \frac{1-a^{M+1}}{1-a}$$

$$\begin{aligned} c_k &= \frac{1}{N}e^{j\frac{2\pi}{N}N_1k}\sum_{m=0}^{2N_1}\left(e^{-(j\frac{2\pi}{N}k)}\right)^m \\ &= \frac{1}{N}e^{j\frac{2\pi}{N}N_1k}\frac{1-e^{-(j\frac{2\pi}{N}(2N_1+1)}\right)}{1-e^{-(jk\frac{2\pi}{N})}} \end{aligned} \tag{7.2.4}$$

Now, using the "partial summation formula"

$$\sum_{n=0}^{M} a^n = \frac{1-a^{M+1}}{1-a}$$

$$\begin{aligned} c_k &= \frac{1}{N}e^{j\frac{2\pi}{N}N_1k}\sum_{m=0}^{2N_1}\left(e^{-(j\frac{2\pi}{N}k)}\right)^m \\ &= \frac{1}{N}e^{j\frac{2\pi}{N}N_1k}\frac{1-e^{-(j\frac{2\pi}{N}(2N_1+1)}\right)}{1-e^{-(jk\frac{2\pi}{N})}} \end{aligned} \tag{7.2.5}$$

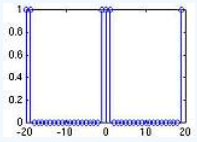
Manipulate to make this look like a sinc function (distribute):

$$\begin{aligned} c_k &= \frac{1}{N}\frac{e^{-(jk\frac{2\pi}{2N})}\left(e^{jk\frac{2\pi}{N}(N_1+\frac{1}{2})} - e^{-(jk\frac{2\pi}{N}(N_1+\frac{1}{2}))}\right)}{e^{-(jk\frac{2\pi}{2N})}\left(e^{jk\frac{2\pi}{N}\frac{1}{2}} - e^{-(jk\frac{2\pi}{N}\frac{1}{2})}\right)} \\ &= \frac{1}{N}\frac{\sin\left(\frac{2\pi k(N_1+\frac{1}{2})}{N}\right)}{\sin\left(\frac{\pi k}{N}\right)} \\ &= \text{digital sinc} \end{aligned} \tag{7.2.6}$$

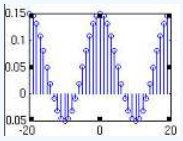
https://eng.libretexts.org/@go/page/22880

Note

It's periodic! Figure 7.2.9, Figure 7.2.10, and Figure 7.2.11 how our above function and coefficients for various values of N_1.

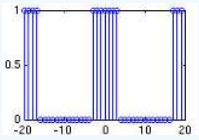


(a)

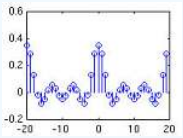


(b)

Figure 7.2.9: $N_1 = 1$ (a) Plot of $f[n]$. (b) Plot of $c[k]$

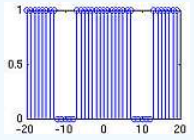


(a)

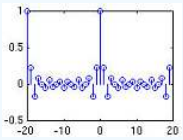


(b)

Figure 7.2.10: $N_1 = 3$ (a) Plot of $f[n]$. (b) Plot of $c[k]$.



(a)



(b)

Figure 7.2.11: $N_1 = 7$ (a) Plot of $f[n]$. (b) Plot of $c[k]$.

DTFS conclusion

Using the steps shown above in the derivation and our previous understanding of Hilbert Spaces (Section 15.4) and Orthogonal Expansions (Section 15.9), the rest of the derivation is automatic. Given a discrete-time, periodic signal (vector in $\mathbb{C}^n$) $f[n]$, we can write:

$$f[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{w}kn}$$

$$c_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f[n] e^{-(j\frac{2\pi}{N}kn)}$$

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:

$$f[n] = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{N}kn}$$

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} f[n] e^{-(j\frac{2\pi}{N}kn)}$$

This is what the `fft` command in MATLAB does.

This page titled 7.2: Discrete Time Fourier Series (DTFS) is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

7.3: Common Discrete Fourier Series

Introduction

Once one has obtained a solid understanding of the fundamentals of **Fourier series analysis** and the **General Derivation of the Fourier Coefficients**, it is useful to have an understanding of the common signals used in Fourier Series Signal Approximation.

Deriving the Coefficients

Consider a square wave $f(x)$ of length 1. Over the range [0,1), this can be written as

$$x(t) = \begin{cases} 1 & t \leq \frac{1}{2} \\ -1 & t > \frac{1}{2} \end{cases}$$

Fourier series approximation of a square wave

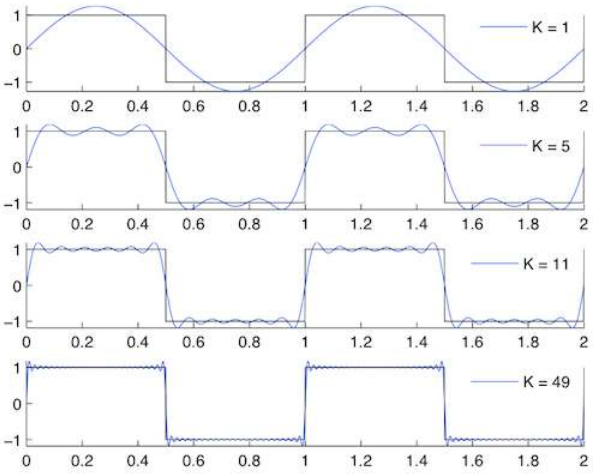


Figure 7.3.1: Fourier series approximation to $sq(t)$. The number of terms in the Fourier sum is indicated in each plot, and the square wave is shown as a dashed line over two periods.

Real Even Signals Given that the square wave is a real and even signal,

- $f(t) = f(-t)$ EVEN
- $f(t) = f^*(t)$ REAL

therefore,

- $c_n = c_{-n}$ EVEN
- $c_n = c_n^*$ REAL

Deriving the Coefficients for other signals

The Square wave is the standard example, but other important signals are also useful to analyze, and these are included here.

Constant Waveform

This signal is relatively self-explanatory: the time-varying portion of the Fourier Coefficient is taken out, and we are left simply with a constant function over all time.

$$x(t) = 1$$

Fourier series approximation of a constant wave

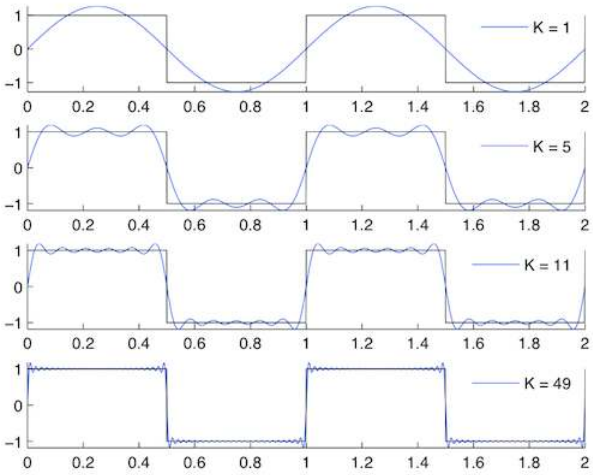


Figure 7.3.2

Sinusoid Waveform

With this signal, only a specific frequency of time-varying Coefficient is chosen (given that the Fourier Series equation includes a sine wave, this is intuitive), and all others are filtered out, and this single time-varying coefficient will exactly match the desired signal.

$$x(t) = \cos(2\pi t)$$

Fourier series approximation of a sinusoid wave

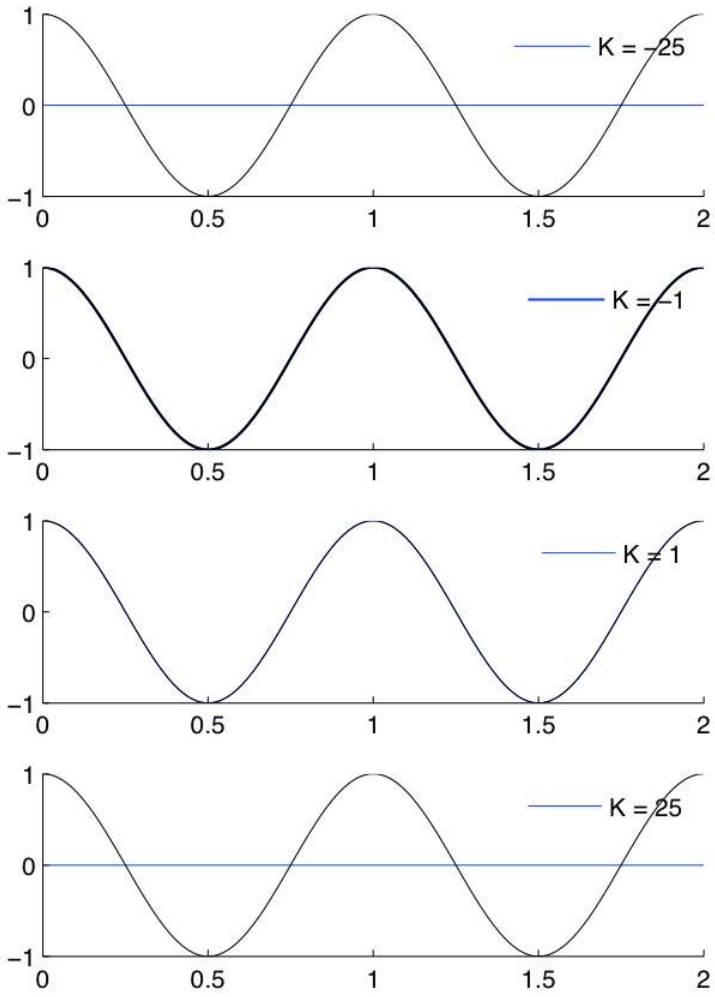


Figure 7.3.3

Triangle Waveform

$$x(t) = \begin{cases} t & t \leq 1/2 \\ 1-t & t > 1/2 \end{cases}$$

This is a more complex form of signal approximation to the square wave. Because of the **Symmetry Properties** of the Fourier Series, the triangle wave is a real and odd signal, as opposed to the real and even square wave signal. This means that

- $f(t) = -f(-t)$ ODD
- $f(t) = f^*(t)$ REAL

therefore,

- $c_n = -c_{-n}$
- $c_n = -c_n^*$ IMAGINARY

Fourier series approximation of a triangle wave

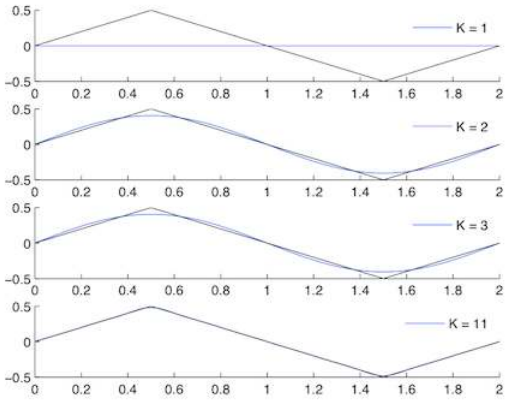


Figure 7.3.4

Sawtooth Waveform

$$x(t) = t/2$$

Because of the **Symmetry Properties** of the Fourier Series, the sawtooth wave can be defined as a real and odd signal, as opposed to the real and even square wave signal. This has important implications for the Fourier Coefficients.

Fourier series approximation of a sawtooth wave

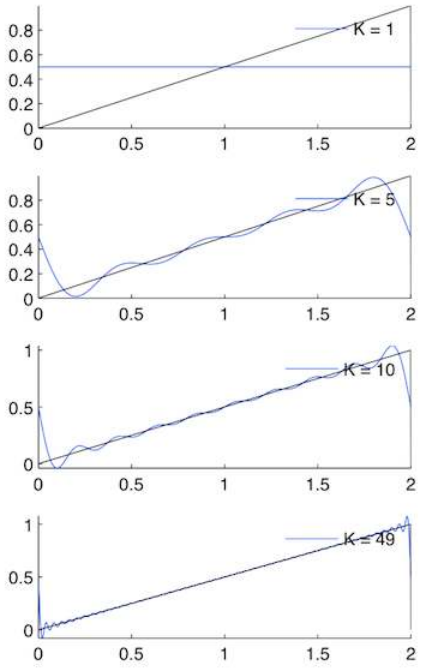


Figure 7.3.5

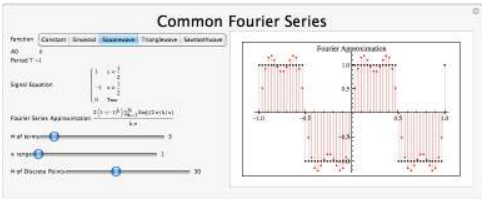


Figure 7.3.6: Interact (when online) with a Mathematica CDF demonstrating the common Discrete Fourier Series. To download, right-click and save as .cdf.

Conclusion

To summarize, a great deal of variety exists among the common Fourier Transforms. A summary table is provided here with the essential information.

Table 7.3.1: Common Discrete Fourier Transforms

Description	Time Domain Signal for $n \in \mathbb{Z}[0, N-1]$	Frequency Domain Signal $k \in \mathbb{Z}[0, N-1]$
Constant Function	1	$\delta(k)$
Unit Impulse	$\delta(n)$	$\frac{1}{N}$
Complex Exponential	$e^{j2\pi mn/N}$	$\delta\left((k-m)_N\right)$
Sinusoid Waveform	$\cos(j2\pi mn/N)$	$\frac{1}{2}\left(\delta\left((k-m)_N\right)+\delta\left((k+m)_N\right)\right)$
Box Waveform $(M < N/2)$	$\delta(n)+\sum_{m=1}^{M}\delta\left((n-m)_N\right)+\delta\left((n+m)_N\right)$	$\frac{\sin((2M+1)k\pi/N)}{N\sin(k\pi/N)}$
Dsinc Waveform $(M < N/2)$	$\frac{\sin((2M+1)n\pi/N)}{\sin(n\pi/N)}$	$\delta(k)+\sum_{m=1}^{M}\delta\left((k-m)_N\right)+\delta((k+m)_N)$

This page titled 7.3: Common Discrete Fourier Series is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

7.4: Properties of the DTFS

Introduction

In this module we will discuss the basic properties of the Discrete-Time Fourier Series. We will begin by refreshing your memory of our basic Fourier series equations:

$$f[n] = \sum_{k=0}^{N-1} c_k e^{j\omega_0 kn}$$

$$c_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f[n] e^{-(j\frac{2\pi}{N}kn)}$$

Let $\mathscr{F}(\cdot)$ denote the transformation from $f[n]$ to the Fourier coefficients

$$\mathscr{F}(f[n]) = c_k \quad , \quad k \in \mathbb{Z}$$

$\mathscr{F}(\cdot)$ maps complex valued functions to sequences of complex numbers.

Linearity

$\mathscr{F}(\cdot)$ is a **linear transformation**.

Theorem 7.4.1

If $\mathscr{F}(f[n]) = c_k$ and $\mathscr{F}(g[n]) = d_k$. Then

$$\mathscr{F}(\alpha f[n]) = \alpha c_k \quad , \quad \alpha \in \mathbb{C}$$

and

$$\mathscr{F}(f[n] + g[n]) = c_k + d_k$$

Proof

Easy. Just linearity of integral.

$$\begin{aligned} \mathscr{F}(f[n] + g[n]) &= \sum_{n=0}^{N} (f[n] + g[n]) e^{-(j\omega_0 kn)}, k \in \mathbb{Z} \\ &= \frac{1}{N} \sum_{n=0}^{N} f[n] e^{-(j\omega_0 kn)} + \frac{1}{N} \sum_{n=0}^{N} g[n] e^{-(j\omega_0 kn)}, k \in \mathbb{Z} \\ &= c_k + d_k, k \in \mathbb{Z} \\ &= c_k + d_k \end{aligned} \tag{7.4.1}$$

Shifting

Shifting in time equals a phase shift of Fourier coefficients.

Theorem 7.4.2

$\mathscr{F}\left(f\left[n - n_0\right]\right) = e^{-(j\omega_0 kn_0)} c_k$ if $c_k = |c_k| e^{j\angle(c_k)}$, then

$$\left| e^{-(j\omega_0 kn_0)} c_k \right| = \left| e^{-(j\omega_0 kn_0)} \right| |c_k| = |c_k|$$

$$\angle\left(e^{-(j\omega_0 n_0 k)}\right) = \angle\left(c_k\right) - \omega_0 n_0 k$$

Proof

$$\begin{aligned}\mathcal{F}\left(f\left[n-n_0\right]\right) &= \frac{1}{N}\sum_{n=0}^{N} f\left[n-n_0\right] e^{-(j\omega_0 kn)}, \quad k\in\mathbb{Z} \\ &= \frac{1}{N}\sum_{n=-n_0}^{N-n_0} f\left[n-n_0\right] e^{-(j\omega_0 k(n-n_0))} e^{-(j\omega_0 k n_0)} \quad , \quad k\in\mathbb{Z} \\ &= \frac{1}{N}\sum_{n=-n_0}^{N-n_0} f[\tilde{n}] e^{-(j\omega_0 k\tilde{n})} e^{-(j\omega_0 k n_0)} \quad , \quad k\in\mathbb{Z} \\ &= e^{-(j\omega_0 k\tilde{n})} c_k \quad , \quad k\in\mathbb{Z} \end{aligned} \tag{7.4.2}$$

Parseval's Relation

$$\sum_{n=0}^{N} \left(|f[n]|\right)^2 = N\sum_{k=0}^{N-1} \left(|c_k|\right)^2$$

Parseval's relation tells us that the energy of a signal is equal to the energy of its Fourier transform.

Note

Parseval tells us that the Fourier series **maps** $L^2[0,N]$ to l^2.

$$f[n] \xleftrightarrow{\text{DTFS}} \{c_k\}$$

$$f[n]\in L^2[0,N] \Longleftrightarrow \{c_k\}\in l^2\{\mathbb{Z}\}$$

Figure 7.4.1

Exercise 7.4.1

For $f[n]$ to have "finite energy," what do the c_k do as $k\to\infty$?

Answer

$(|c_k|)^2 < \infty$ for $f[n]$ to have finite energy.

Exercise 7.4.2

If $\forall k, |k|>0 : (c_k = \frac{1}{k})$, is $f\in L^2[[0,N]]$?

Answer

Yes, because $(|c_k|)^2 = \frac{1}{k^2}$, which is summable.

Exercise 7.4.3

Now, if $\forall k, |k|>0 : \left(c_k = \frac{1}{\sqrt{k}}\right)$, is $f\in L^2[[0,N]]$?

Answer

No, because $(|c_k|)^2 = \frac{1}{k}$, which is not summable.

The rate of decay of the Fourier series determines if $f[n]$ has **finite energy**.

ParsevalsTheorem Demonstration

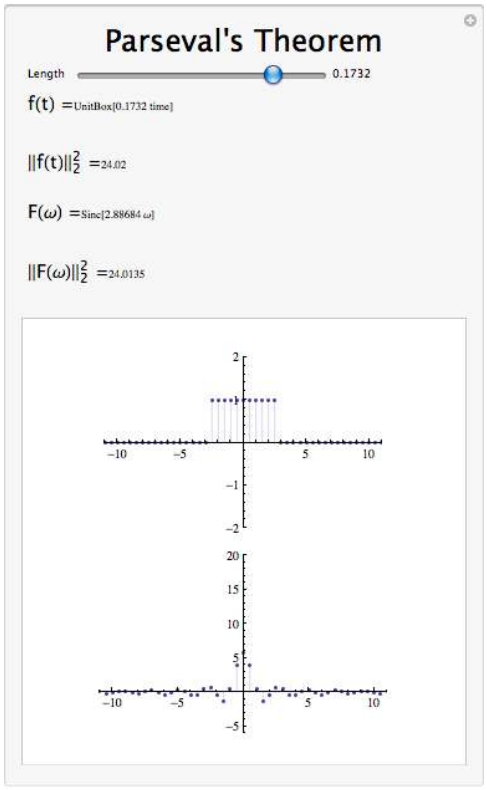


Figure 7.4.2: Interact (when online) with a Mathematica CDF demonstrating Parsevals Theorem. To download, right-click and save as .cdf.

Symmetry Properties

Rule 7.4.1: Even Signals

$f[n] = f[-n]$

$\|c_k\| = \|c_{-k}\|$

Proof

$$\begin{aligned} c_k =& \frac{1}{N}\sum_{0}^{N}[n]\exp\left[-j\omega_0 kn\right] \\ =& \frac{1}{N}\sum_{0}^{\frac{N}{2}} f[n]\exp\left[-j\omega_0 kn\right] + \frac{1}{N}\sum_{\frac{N}{2}}^{N} f[n]\exp\left[-j\omega_0 kn\right] \\ =& \frac{1}{N}\sum_{0}^{\frac{N}{2}} f[-n]\exp\left[-j\omega_0 kn\right] + \frac{1}{N}\sum_{\frac{N}{2}}^{N} f[-n]\exp\left[-j\omega_0 kn\right] \\ =& \frac{1}{N}\sum_{0}^{N}[n]\left[\exp\left[j\omega_0 kn\right] + \exp\left[-j\omega_0 kn\right]\right] \\ =& \frac{1}{N}\sum_{0}^{N}[n]2\cos\left[\omega_0 kn\right] \end{aligned} \tag{7.4.3}$$

Rule 7.4.2: Odd Signals

$f[n] = -f[-n]$

$c_k = c_{-k}^*$

Proof

$$\begin{aligned} c_k =& \frac{1}{N}\sum_{0}^{N} f[n]\exp\left[-\mathrm{j}\omega_0 kn\right] \\ =& \frac{1}{N}\sum_{0}^{\frac{N}{2}} f[n]\exp\left[-\mathrm{j}\omega_0 kn\right] + \frac{1}{N}\sum_{\frac{N}{2}}^{N} f[n]\exp\left[-\mathrm{j}\omega_0 kn\right] \\ =& \frac{1}{N}\sum_{0}^{\frac{N}{2}} f[n]\exp\left[-\mathrm{j}\omega_0 kn\right] - \frac{1}{N}\sum_{\frac{N}{2}}^{N} f[-n]\exp\left[\mathrm{j}\omega_0 kn\right] \\ =& -\frac{1}{N}\sum_{0}^{N} f[n]\left[\exp\left[\mathrm{j}\omega_0 kn\right] - \exp\left[-\mathrm{j}\omega_0 kn\right]\right] \\ =& -\frac{1}{N}\sum_{0}^{N} f[n]2\,\mathrm{j}\sin\left[\omega_0 kn\right] \end{aligned} \tag{7.4.4}$$

Rule 7.4.3: Real Signals

$f[n] = f^*[n]$

$c_k = c_{-k}^*$

Proof

Erroneous nesting of equation structures

Differentiation in Fourier Domain

$$\left(\mathscr{F}(f[n]) = c_k\right) \Rightarrow \left(\mathscr{F}\left(\frac{\mathrm{d}f[n]}{\mathrm{d}n}\right) = jk\omega_0 c_k\right)$$

Since

$$f[n] = \sum_{i=0}^{N} c_k e^{j\omega_0 kn}$$

then

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{dn}} f[n] &= \sum_{k=0}^{N} c_k \frac{\mathrm{d}e^{j\omega_0 kn}}{\mathrm{dn}} \\ &= \sum_{k=0}^{N} c_k j\omega_0 k e^{j\omega_0 kn} \end{aligned} \tag{7.4.5}$$

A differentiator **attenuates** the low frequencies in $f[n]$ 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 $f[n]$ 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 $k \to \infty$. If $\mathscr{F}(f[n]) = c_k$ and $|c_k|$ has the form $\frac{1}{k^l}$, then $\mathscr{F}\left(\frac{\mathrm{d}^m f[n]}{\mathrm{d}n^m}\right) = (jk\omega_0)^m c_k$ and has the form $\frac{k^m}{k^l}$. So for the m^{th} derivative to have finite energy, we need

$$\sum_k \left(\left|\frac{k^m}{k^l}\right|\right)^2 < \infty$$

thus $\frac{k^m}{k^l}$ decays **faster** than $\frac{1}{k}$ which implies that

$$2l - 2m > 1$$

or

$$l > \frac{2m+1}{2}$$

Thus the decay rate of the Fourier series dictates smoothness.

Fourier Differentiation Demo

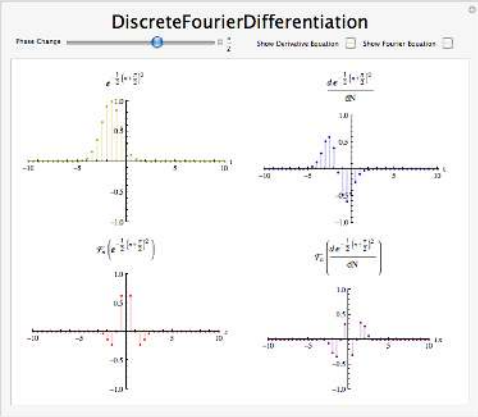


Figure 7.4.3: Interact (when online) with a Mathematica CDF demonstrating Differentiation in a Fourier Domain. To download, right-click and save as .cdf.

Integration in the Fourier Domain

If

$$\mathscr{F}(f[n]) = c_k$$

then

$$\mathscr{F}\left(\sum_{\eta=0}^{n} f[\eta]\right) = \frac{1}{j\omega_0 k} c_k$$

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.

Fourier Integration Demo

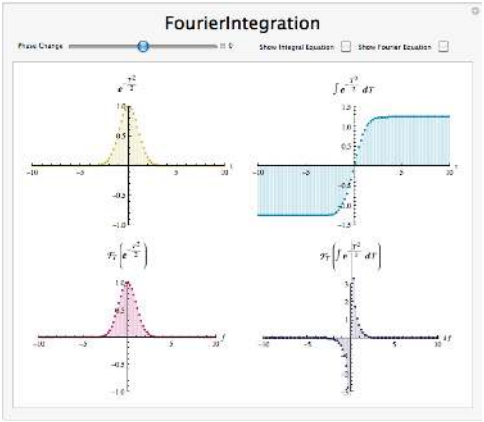


Figure 7.4.4: Interact (when online) with a Mathematica CDF demonstrating the effect of Integration in a Fourier Domain.To download, right-click and save as .cdf.

Signal Multiplication

Given a signal $f[n]$ with Fourier coefficients c_k and a signal $g[n]$ with Fourier coefficients d_k, we can define a new signal, $y[n]$, where $y[n] = f[n]g[n]$. We find that the Fourier Series representation of $y[n]$, e_k, is such that $e_k = \sum_{l=0}^{N} c_l d_{k-l}$. This is to say that signal multiplication in the time domain is equivalent to discrete-time circular convolution (Section 4.3) in the frequency domain. The proof of this is as follows

$$
\begin{aligned}
e_k &= \frac{1}{N}\sum_{n=0}^{N} f[n]g[n]e^{-(j\omega_0 kn)} \\
&= \frac{1}{N}\sum_{n=0}^{N}\sum_{l=0}^{N} c_l e^{j\omega_0 ln} g[n]e^{-(j\omega_0 kn)} \\
&= \sum_{l=0}^{N} c_l \left(\frac{1}{N}\sum_{n=0}^{N} g[n]e^{-(j\omega_0 (k-l)n)}\right) \\
&= \sum_{l=0}^{N} c_l d_{k-l}
\end{aligned} \tag{7.4.6}
$$

Conclusion

Like other Fourier transforms, the DTFS has many useful properties, including linearity, equal energy in the time and frequency domains, and analogs for shifting, differentation, and integration.

Table 7.4.1: Properties of the Discrete Fourier Transform

Property	Signal	DTFS
Linearity	$ax(n) + by(n)$	$aX(k) + bY(k)$
Time Shifting	$x(n-m)$	$X(k)e^{-j2\pi mk/N}$
Time Modulation	$x(n)e^{j2\pi mn/N}$	$X(k-m)$
Multiplication	$x(n)y(n)$	$X(k) * Y(k)$
Circular Convolution	$x(n) * y(n)$	$X(k)Y(K)$

This page titled 7.4: Properties of the DTFS is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

7.5: Discrete Time Circular Convolution and the DTFS

Introduction

This module relates circular convolution of periodic signals in one domain to multiplication in the other domain.

You should be familiar with Discrete-Time Convolution (Section 4.3), 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

$$\begin{aligned} y[n] &= x[n] * h[n] \\ &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \end{aligned} \tag{7.5.1}$$

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 $n = 0, 1, \ldots, N-1$. This idea led to the development of **circular convolution**, also called cyclic or periodic convolution.

Signal Circular Convolution

Given a signal $f[n]$ with Fourier coefficients c_k and a signal $g[n]$ with Fourier coefficients d_k, we can define a new signal, $v[n]$, where $v[n] = f[n] \circledast g[n]$. We find that the Fourier Series representation of $v[n]$, a_k, is such that $a_k = c_k d_k$. $f[n] \circledast g[n]$ is the circular convolution (Section 7.5) of two periodic signals and is equivalent to the convolution over one interval, i.e. $f[n] \circledast g[n] = \sum_{n=0}^{N} \sum_{\eta=0}^{N} f[\eta] g[n-\eta]$.

Note

Circular convolution in the time domain is equivalent to multiplication of the Fourier coefficients.

This is proved as follows

$$\begin{aligned} a_k &= \frac{1}{N} \sum_{n=0}^{N} v[n] e^{-(j\omega_0 kn)} \\ &= \frac{1}{N^2} \sum_{n=0}^{N} \sum_{n=0}^{\eta} f[\eta] g[n-\eta] e^{-(j\omega_0 kn)} \\ &= \frac{1}{N} \sum_{\eta=0}^{N} f[\eta] \left(\frac{1}{N} \sum_{n=0}^{N} g[n-\eta] e^{-(j\omega_0 kn)} \right) \\ &= \left(\frac{1}{N} \sum_{\eta=0}^{N} f[\eta] \left(\frac{1}{N} \sum_{\nu=-\eta}^{N-\eta} g[\nu] e^{-(j\omega_0 k(\nu+\eta))} \right) \right) \quad , \quad \nu = n - \eta \\ &= \frac{1}{N} \sum_{\eta=0}^{N} f[\eta] \left(\frac{1}{N} \sum_{\nu=-\eta}^{N-\eta} g[\nu] e^{-(j\omega_0 k\nu)} \right) e^{-(j\omega_0 k\eta)} \\ &= \frac{1}{N} \sum_{\eta=0}^{N} f[\eta] d_k e^{-(j\omega_0 k\eta)} \\ &= d_k \left(\frac{1}{N} \sum_{\eta=0}^{N} f[\eta] e^{-(j\omega_0 k\eta)} \right) \\ &= c_k d_k \end{aligned} \tag{7.5.2}$$

Circular Convolution Formula

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

$$Y[k] = F[k]H[k]$$

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

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

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

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

$$\begin{aligned} y[n] &= \frac{1}{N}\sum_{k=0}^{N-1}\sum_{m=0}^{N-1} f[m]e^{(-j)\frac{2\pi}{N}kn}H[k]e^{j\frac{2\pi}{N}kn} \\ &= \sum_{m=0}^{N-1} f[m]\left(\frac{1}{N}\sum_{k=0}^{N-1} H[k]e^{j\frac{2\pi}{N}k(n-m)}\right) \end{aligned} \tag{7.5.3}$$

where we can reduce the second summation found in the above equation into $h\left[((n-m))_N\right] = \frac{1}{N}\sum_{k=0}^{N-1} H[k]e^{j\frac{2\pi}{N}k(n-m)}y[n] = \sum_{m=0}^{N-1} f[m]h\left[((n-m))_N\right]$ which equals circular convolution! When we have $0 \leq n \leq N-1$ in the above, then we get:

$$y[n] \equiv f[n] \circledast h[n]$$

Note

The notation $\circledast$ represents cyclic convolution "mod N".

Alternative Convolution Formula

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: $y[n] = \sum_{m=0}^{N-1} f[m]h\left[((n-m))_N\right]$. For each n : N multiples, $N-1$ additions.

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

Steps for Circular Convolution

We can picture periodic (Section 6.1) sequences as having discrete points on a circle as the domain

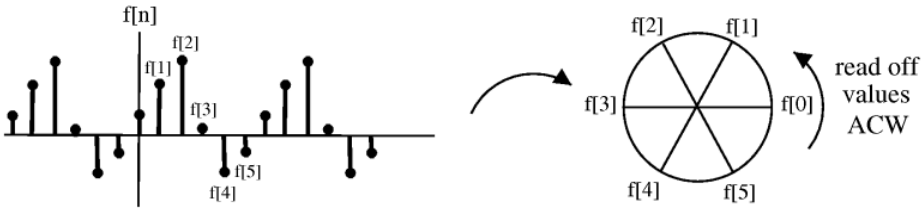


Figure 7.5.1

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:

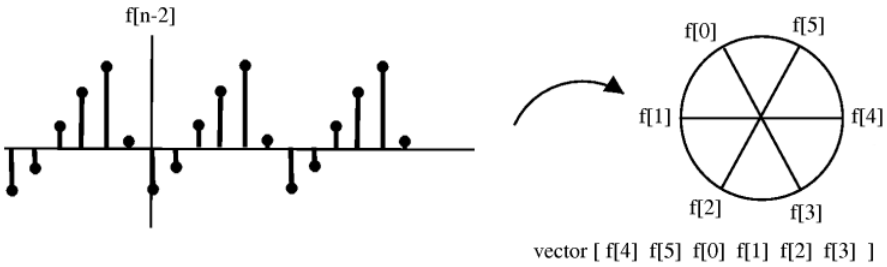


Figure 7.5.2: for $m = -2$

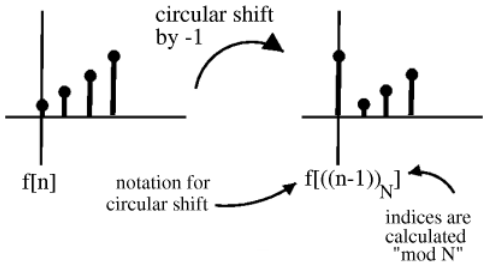


Figure 7.5.3

To cyclic shift we follow these steps:

1) Write $f(n)$ on a cylinder, ACW

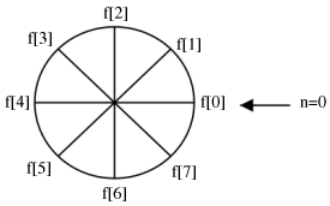


Figure 7.5.4: $N = 8$

2) To cyclic shift by m, spin cylinder m spots ACW

$$f[n] \rightarrow f\left[((n+m))_N\right]$$

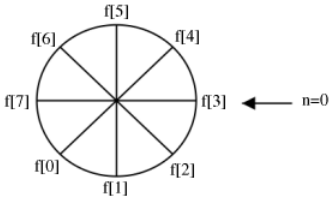


Figure 7.5.5: $m = -3$

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 mm is equivalent to shifting CW $N - m$

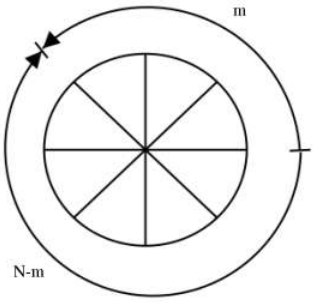
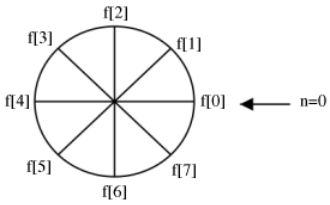
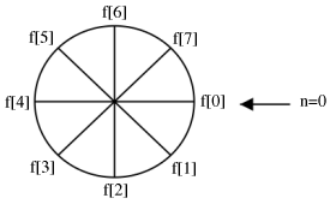


Figure 7.5.6

$f[((-n))_N]$ The above expression, simply writes the values of $f[n]$ clockwise.



(a) $f[n]$

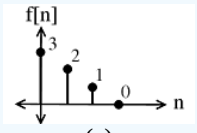


(b) $f[((-n))_N]$

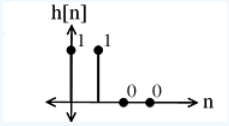
Figure 7.5.7

Example 7.5.1

Convolve (n = 4)



(a)



(b)

Figure 7.5.8: Two discrete-time signals to be convolved.

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

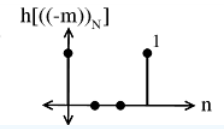


Figure 7.5.9

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

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

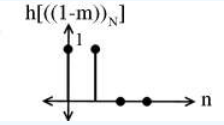


Figure 7.5.10

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

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

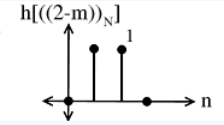


Figure 7.5.11

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

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

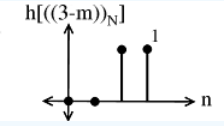


Figure 7.5.12

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

Exercise

Take a look at a square pulse with a period of T.

For this signal $c_k = \begin{cases} \frac{1}{N} \text{ if } k = 0 \\ \frac{1}{2}\frac{\sin(\frac{\pi}{2}k)}{\frac{\pi}{2}k} \text{ otherwise} \end{cases}$

Take a look at a triangle pulse train with a period of T.

This signal is created by circularly convolving the square pulse with itself. The Fourier coefficients for this signal are $a_k = c_k^2 = \frac{1}{4}\frac{\sin^2}{(\frac{\pi}{2}k)}$

Exercise 7.5.1

Find the Fourier coefficients of the signal that is created when the square pulse and the triangle pulse are convolved.

Answer

$$a_k = \begin{cases} \text{undefined} & k = 0 \\ \frac{1}{8}\frac{\sin^3[\frac{\pi}{2}k]}{[\frac{\pi}{2}k]^3} & \text{otherwise} \end{cases}$$

Circular Shifts and the DFT

Theorem 7.5.1: Circular Shifts and DFT

If $f[n] \overset{\text{DFT}}{\longleftrightarrow} F[k]$ then $f\left[((n-m))_N\right] \overset{\text{DFT}}{\longleftrightarrow} e^{-(j\frac{2\pi}{N}km)}F[k]$ (i.e. circular shift in time domain = phase shift in DFT)

Proof

$$f[n] = \frac{1}{N}\sum_{k=0}^{N-1} F[k]e^{j\frac{2\pi}{N}kn}$$

so phase shifting the DFT

$$\begin{aligned} f[n] &= \frac{1}{N}\sum_{k=0}^{N-1} F[k]e^{-(j\frac{2\pi}{N}kn)}e^{j\frac{2\pi}{N}kn} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} F[k]e^{j\frac{2\pi}{N}k(n-m)} \\ &= f\left[((n-m))_N\right] \end{aligned} \tag{7.5.4}$$

Circular Convolution Demonstration

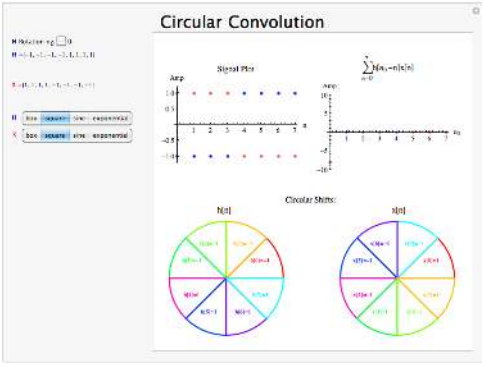


Figure 7.5.13: Interact (when online) with a Mathematica CDF demonstrating Circular Shifts.

Conclusion

Circular convolution in the time domain is equivalent to multiplication of the Fourier coefficients in the frequency domain.

This page titled 7.5: Discrete Time Circular Convolution and the DTFS is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

8: Continuous Time Fourier Transform (CTFT)

8.1: Continuous Time Aperiodic Signals

8.2: Continuous Time Fourier Transform (CTFT)

8.3: Common Fourier Transforms

8.4: Properties of the CTFT

8.5: Continuous Time Convolution and the CTFT

This page titled 8: Continuous Time Fourier Transform (CTFT) is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

8.1: Continuous Time Aperiodic Signals

Introduction

This module describes the type of signals acted on by the Continuous Time Fourier Transform.

Relevant Spaces

The Continuous-Time Fourier Transform maps infinite-length (a-periodic), continuous-time signals in L^2 to infinite-length, discrete-frequency signals in l^2.

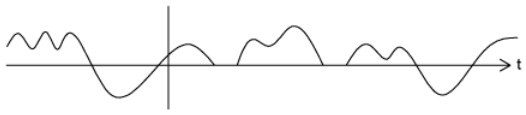


Figure 8.1.1

Periodic and Aperiodic Signals

When a function repeats itself exactly after some given period, or cycle, we say it's **periodic**. A **periodic function** can be mathematically defined as:

$$f(t) = f(t + mT) \quad m \in \mathbb{Z} \tag{8.1.1}$$

where $T > 0$ represents the **fundamental period** of the signal, which is the smallest positive value of T for the signal to repeat. Because of this, you may also see a signal referred to as a T-periodic signal. Any function that satisfies this equation is said to be **periodic** with period T.

An **aperiodic** CT function $f(t)$ does not repeat for any $T \in \mathbb{R}$; i.e. there exists no T such that Equation 8.1.1 holds.

Suppose we have such an aperiodic function $f(t)$. We can construct a periodic extension of $f(t)$ called $f_{T_o}(t)$, where $f(t)$ is repeated every T_0 seconds. If we take the limit as $T_0 \to \infty$, we obtain a precise model of an aperiodic signal for which all rules that govern periodic signals can be applied, including Fourier Analysis (with an important modification). For more detail on this distinction, see the module on the **Continuous Time Fourier Transform**.

Aperiodic Signal Demonstration

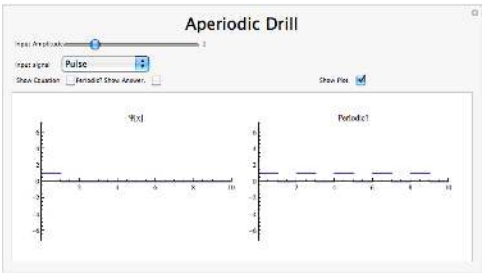


Figure $PageIndex2$: Interact (when online) with a Mathematica CDF demonstrating Periodic versus Aperiodic Signals.To download, right-click and save as .cdf.

Conclusion

Any aperiodic signal can be defined by an infinite sum of periodic functions, a useful definition that makes it possible to use Fourier Analysis on it by assuming all frequencies are present in the signal.

This page titled 8.1: Continuous Time Aperiodic Signals is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

8.2: Continuous Time Fourier Transform (CTFT)

Introduction

In this module, we will derive an expansion for any arbitrary continuous-time function, and in doing so, derive the **Continuous Time Fourier Transform** (CTFT).

Since complex exponentials (Section 1.8) are eigenfunctions of linear time-invariant (LTI) systems (Section 14.5), calculating the output of an LTI system $\mathscr{H}$ given e^{st} as an input amounts to simple multiplication, where $H(s) \in \mathbb{C}$ is the eigenvalue corresponding to s. As shown in the figure, a simple exponential input would yield the output

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

Using this and the fact that $\mathscr{H}$ is linear, calculating $y(t)$ for combinations of complex exponentials is also straightforward.

$$c_1 e^{s_1 t} + c_2 e^{s_2 t} \rightarrow c_1 H\left(s_1\right) e^{s_1 t} + c_2 H\left(s_2\right) e^{s_2 t}$$
$$\sum_n c_n e^{s_n t} \rightarrow \sum_n c_n H\left(s_n\right) e^{s_n t}$$

The action of H on an input such as those in the two equations above is easy to explain. $\mathscr{H}$ **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 a system.

Now, we will look to use the power of complex exponentials to see how we may represent arbitrary signals in terms of a set of simpler functions by superposition of a number of complex exponentials. Below we will present the **Continuous-Time Fourier Transform** (CTFT), commonly referred to as just the Fourier Transform (FT). Because the CTFT deals with nonperiodic signals, we must find a way to include all real frequencies in the general equations. For the CTFT we simply utilize integration over real numbers rather than summation over integers in order to express the aperiodic signals.

Fourier Transform Synthesis

Joseph Fourier demonstrated that an arbitrary $s(t)$ can be written as a linear combination of harmonic complex sinusoids

$$s(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\omega_0 nt} \tag{8.2.1}$$

where $\omega_0 = \frac{2\pi}{T}$ is the fundamental frequency. For almost all $s(t)$ of practical interest, there exists c_n to make Equation 8.2.1 true. If $s(t)$ is finite energy ($s(t) \in L^2\ [0,T]$), then the equality in Equation 8.2.1 holds in the sense of energy convergence; if $s(t)$ is continuous, then Equation 8.2.1 holds pointwise. Also, if $s(t)$ meets some mild conditions (the Dirichlet conditions), then Equation 8.2.1 holds pointwise everywhere except at points of discontinuity.

The c_n - called the Fourier coefficients - tell us "how much" of the sinusoid $e^{j\omega_0 nt}$ is in $s(t)$. The formula shows $s(t)$ as a sum of complex exponentials, each of which is easily processed by an LTI system (since it is an eigenfunction of **every** LTI system). Mathematically, it tells us that the set of complex exponentials $\left\{e^{j\omega_0 nt}, \quad n \in \mathbb{Z}\right\}$ form a basis for the space of T-periodic continuous time functions.

Equations

Now, in order to take this useful tool and apply it to arbitrary non-periodic signals, we will have to delve deeper into the use of the superposition principle. Let $s_T(t)$ be a periodic signal having period T. We want to consider what happens to this signal's spectrum as the period goes to infinity. We denote the spectrum for any assumed value of the period by $c_n(T)$. We calculate the spectrum according to the Fourier formula for a periodic signal, known as the Fourier Series (for more on this derivation, see the section on **Fourier Series**.)

$$c_n = \frac{1}{T}\int_0^T s(t)\exp\left(-j\omega_0 t\right)dt$$

where $\omega_0 = \frac{2\pi}{T}$ and where we have used a symmetric placement of the integration interval about the origin for subsequent derivational convenience. We vary the frequency index nn proportionally as we increase the period. Define

$$S_T(f) \equiv Tc_n = \frac{1}{T}\int_0^T S_T(f)\exp\left(j\omega_0 t\right)dt$$

making the corresponding Fourier Series

$$s_T(t) = \sum_{-\infty}^{\infty} f(t)\exp\left(j\omega_0 t\right)\frac{1}{T}$$

As the period increases, the spectral lines become closer together, becoming a continuum. Therefore,

$$\lim_{T\to\infty} s_T(t) \equiv s(t) = \int_{-\infty}^{\infty} S(f)\exp\left(j\omega_0 t\right)df$$

with

$$S(f) = \int_{-\infty}^{\infty} s(t)\exp\left(-j\omega_0 t\right)dt$$

Continuous-Time Fourier Transform

$$\mathscr{F}(\Omega) = \int_{-\infty}^{\infty} f(t)e^{-(j\Omega t)}\mathrm{d}t \tag{8.2.2}$$

Inverse CTFT

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathscr{F}(\Omega)e^{j\Omega t}\mathrm{d}\Omega \tag{8.2.3}$$

Note

It is not uncommon to see the above formula written slightly different. One of the most common differences is the way that the exponential is written. The above equations use the radial frequency variable Ω in the exponential, where $\Omega = 2\pi f$, but it is also common to include the more explicit expression, $j2\pi ft$, in the exponential. Click here for an overview of the notation used in Connexion's DSP modules.

We know from Euler's formula that $\cos(\omega t) + \sin(\omega t) = \frac{1-j}{2}e^{j\omega t} + \frac{1+j}{2}e^{-j\omega t}$.

CTFT Definition Demonstration

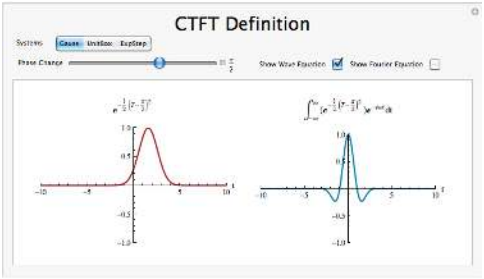


Figure 8.2.1: Interact (when online) with a Mathematica CDF demonstrating Continuous Time Fourier Transform. To Download, right-click and save as .cdf.

Example Problems

Exercise 8.2.1

Find the Fourier Transform (CTFT) of the function

$$f(t) = \begin{cases} e^{-(\alpha t)} \text{ if } t \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

Answer

In order to calculate the Fourier transform, all we need to use is Equation 8.2.2, complex exponentials (Section 1.8), and basic calculus.

$$\begin{aligned}\mathcal{F}(\Omega) &= \int_{-\infty}^{\infty} f(t)e^{-(j\Omega t)}\mathrm{d}t \\ &= \int_{0}^{\infty} e^{-(at)}e^{-(j\Omega t)}\mathrm{d}t \\ &= \int_{0}^{\infty} e^{(-t)(a+j\Omega)}\mathrm{d}t \\ &= 0 - \frac{-1}{a+j\Omega} \\ \mathcal{F}(\Omega) &= \frac{1}{\alpha+j\Omega} \end{aligned} \tag{8.2.4}$$

Exercise 8.2.2

Find the inverse Fourier transform of the ideal lowpass filter defined by

$$X(\Omega) = \begin{cases} 1 & \text{if } |\Omega| \leq \mathrm{M} \\ 0 & \text{otherwise} \end{cases}$$

Answer

Here we will use Equation 8.2.3 to find the inverse FT given that $t \neq 0$.

$$\begin{aligned} x(t) &= \frac{1}{2\pi}\int_{-M}^{M} e^{j(\Omega,t)}\mathrm{d}\Omega \\ &= \frac{1}{2\pi}e^{j(\Omega,t)}\Big|_{\Omega,\Omega=e^{jw}} \\ &= \frac{1}{\pi t}\sin(Mt) \\ x(t) &= \frac{M}{\pi}\left(\operatorname{sinc}\frac{Mt}{\pi}\right) \end{aligned} \tag{8.2.5}$$

Fourier Transform Summary

Because complex exponentials are eigenfunctions of LTI systems, it is often useful to represent signals using a set of complex exponentials as a basis. The continuous time Fourier series synthesis formula expresses a continuous time, periodic function as the sum of continuous time, discrete frequency complex exponentials.

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

The continuous time Fourier series analysis formula gives the coefficients of the Fourier series expansion.

$$c_n = \frac{1}{T}\int_{0}^{T} f(t)e^{-(j\omega_0 nt)}\mathrm{d}t$$

In both of these equations $\omega_0 = \frac{2\pi}{T}$ is the fundamental frequency.

This page titled 8.2: Continuous Time Fourier Transform (CTFT) is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

8.3: Common Fourier Transforms

Common CTFT Properties

Table 8.3.1

Time Domain Signal	Frequency Domain Signal	Condition
$e^{-(at)}u(t)$	$\frac{1}{a+j\omega}$	$a>0$
$e^{at}u(-t)$	$\frac{1}{a-j\omega}$	$a>0$
$e^{-(a\lvert t\rvert)}$	$\frac{2a}{a^2+\omega^2}$	$a>0$
$te^{-(at)}u(t)$	$\frac{1}{(a+j\omega)^2}$	$a>0$
$t^n e^{-(at)}u(t)$	$\frac{n!}{(a+j\omega)^{n+1}}$	$a>0$
$\delta(t)$	1	
1	$2\pi\delta(\omega)$	
$e^{j\omega_0 t}$	$2\pi\delta\left(\omega-\omega_0\right)$	
$\cos(\omega_0 t)$	$\pi\left(\delta\left(\omega-\omega_0\right)+\delta\left(\omega+\omega_0\right)\right)$	
$\sin(\omega_0 t)$	$j\pi\left(\delta\left(\omega+\omega_0\right)-\delta\left(\omega-\omega_0\right)\right)$	
$u(t)$	$\pi\delta(\omega)+\frac{1}{j\omega}$	
$\operatorname{sgn}(t)$	$\frac{2}{j\omega}$	
$\cos\left(\omega_0 t\right)u(t)$	$\frac{\pi}{2}\left(\delta\left(\omega-\omega_0\right)+\delta\left(\omega+\omega_0\right)\right)+\frac{j\omega}{\omega_0^2-\omega^2}$	
$\sin\left(\omega_0 t\right)u(t)$	$\frac{\pi}{2j}\left(\delta\left(\omega-\omega_0\right)-\delta\left(\omega+\omega_0\right)\right)+\frac{\omega_0}{\omega_0^2-\omega^2}$	
$e^{-(at)}\sin\left(\omega_0 t\right)u(t)$	$\frac{\omega_0}{(a+j\omega)^2+\omega_0^2}$	$a>0$
$e^{-(at)}\cos\left(\omega_0 t\right)u(t)$	$\frac{a+j\omega}{(a+j\omega)^2+\omega_0^2}$	$a>0$
$u(t+\tau)-u(t-\tau)$	$2\tau\frac{\sin(\omega\tau)}{\omega\tau}=2\tau\operatorname{sinc}(\omega t)$	
$\frac{\omega_0}{\pi}\frac{\sin(\omega_0 t)}{\omega_0 t}=\frac{\omega_0}{\pi}\operatorname{sinc}\left(\omega_0\right)$	$u\left(\omega+\omega_0\right)-u\left(\omega-\omega_0\right)$	
$\left(\frac{t}{\tau}+1\right)\left(u\left(\frac{t}{\tau}+1\right)-u\left(\frac{t}{\tau}\right)\right)$ $\left(-\frac{t}{\tau}+1\right)\left(u\left(\frac{t}{\tau}\right)-u\left(\frac{t}{\tau}-1\right)\right)=$ $\operatorname{triag}\left(\frac{t}{2\tau}\right)$	$\tau\operatorname{sinc}^2\left(\frac{\omega\tau}{2}\right)$	
$\frac{\omega_0}{2\pi}\operatorname{sinc}^2\left(\frac{\omega_0 t}{2}\right)$	$\left(\frac{\omega}{\omega_0}+1\right)\left(u\left(\frac{\omega}{\omega_0}+1\right)-u\left(\frac{\omega}{\omega_0}\right)\right)+$ $\left(-\frac{\omega}{\omega_0}+1\right)\left(u\left(\frac{\omega}{\omega_0}\right)-u\left(\frac{\omega}{\omega_0}-1\right)\right)=$ $\operatorname{triag}\left(\frac{\omega}{2\omega_0}\right)$	
$\sum_{n=-\infty}^{\infty}\delta(t-nT)$	$\omega_0\sum_{n=-\infty}^{\infty}\delta\left(\omega-n\omega_0\right)$	$\omega_0=\frac{2\pi}{T}$
$e^{-\frac{t^2}{2\sigma^2}}$	$\sigma\sqrt{2\pi}e^{-\frac{\sigma^2\omega^2}{2}}$	

triag[n] is the triangle function for arbitrary real-valued n.

$$\text{triag[n]}=\begin{cases}1+n & \text{if } -1\le n\le 0\\ 1-n & \text{if } 0<n\le 1\\ 0 & \text{otherwise}\end{cases}$$

This page titled 8.3: Common Fourier Transforms is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

8.4: Properties of the CTFT

Introduction

This module will look at some of the basic properties of the Continuous-Time Fourier Transform (CTFT) (Section 8.2).

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.

Discussion of Fourier Transform Properties

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 8.3) of transforms to find the transform of a more complicated signal.

Example 8.4.1

We will begin with the following signal:

$$z(t) = af_1(t) + bf_2(t)$$

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.

$$Z(\omega) = aF_1(\omega) + bF_2(\omega)$$

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.

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 pulse 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.

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 is proven below:

Example 8.4.2

We will begin by letting $z(t) = f(t-\tau)$. Now let us take the Fourier transform with the previous expression substituted in for $z(t)$.

$$Z(\omega) = \int_{-\infty}^{\infty} f(t-\tau)e^{-(i\omega t)}\mathrm{d}t$$

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.

$$\begin{aligned} Z(\omega) &= \int_{-\infty}^{\infty} f(\sigma)e^{-(i\omega(\sigma+\tau)t)}\mathrm{d}\tau \\ &= e^{-(i\omega\tau)}\int_{-\infty}^{\infty} f(\sigma)e^{-(i\omega\sigma)}\mathrm{d}\sigma \\ &= e^{-(i\omega\tau)}F(\omega) \end{aligned} \tag{8.4.1}$$

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 3.3) module for a more in depth explanation and derivation.

$$\begin{aligned} y(t) &= (f_1(t), f_2(t)) \\ &= \int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)\mathrm{d}\tau \end{aligned} \tag{8.4.2}$$

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 11.1).

Parseval's Relation

$$\int_{-\infty}^{\infty} (|f(t)|)^2\mathrm{d}t = \int_{-\infty}^{\infty} (|F(\omega)|)^2\mathrm{d}f$$

Parseval's relation tells us that the energy of a signal is equal to the energy of its Fourier transform.

$$f(t) \xleftrightarrow{\text{CTFT}} F(\omega)$$

Figure 8.4.1

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; however, here we would use the inverse Fourier transform in place of the Fourier transform. Since we went through the steps in the previous, time-shift proof, below we will just show the initial and final step to this proof:

$$z(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega-\phi)e^{j\omega t}d\omega$$

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:

$$z(t) = f(t)e^{j\phi t}$$

Properties Demonstration

An interactive example demonstration of the properties is included below:

Figure 8.4.2: Interactive Signal Processing Laboratory Virtual Instrument created using NI's Labview.

Summary Table of CTFT Properties

Table 8.4.1: Table of Fourier Transform Properties

Operation Name	Signal ($f(t)$)	Transform ($F(\omega)$)
Linearity	$a\left(f_1, t\right)+b\left(f_2, t\right)$	$a\left(F_1, \omega\right)+b\left(F_2, \omega\right)$
Scalar Multiplication	$\alpha f(t)$	$\alpha F(\omega)$
Symmetry	$F(t)$	$2 \pi f(-\omega)$
Time Scaling	$f(\alpha t)$	$\frac{1}{\lvert\alpha\rvert} F\left(\frac{w}{\alpha}\right)$
Time Shift	$f(t-\tau)$	$F(\omega) e^{-(j \omega \tau)}$
Convolution in Time	$\left(f_1(t), f_2(t)\right)$	$F_1(t) F_2(t)$
Convolution in Frequency	$f_1(t) f_2(t)$	$\frac{1}{2 \pi}\left(F_1(t), F_2(t)\right)$
Differentiation	$\frac{d^n}{d t^n} f(t)$	$(j \omega)^n F(\omega)$
Parseval's Theorem	$\int_{-\infty}^{\infty}\left(\lvert f(t)\rvert\right)^2 d t$	$\int_{-\infty}^{\infty}\left(\lvert F(\omega)\rvert\right)^2 d f$
Modulation (Frequency Shift)	$f(t) e^{j \phi t}$	$F(\omega-\phi)$

This page titled 8.4: Properties of the CTFT is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

8.5: Continuous Time Convolution and the CTFT

Introduction

This module discusses convolution of continuous signals in the time and frequency domains.

Continuous Time Fourier Transform

The CTFT transforms a infinite-length continuous signal in the time domain into an infinite-length continuous signal in the frequency domain.

CTFT

$$\mathcal{F}(\Omega) = \int_{-\infty}^{\infty} f(t) e^{-(j\Omega t)} dt$$

Inverse CTFT

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}(\Omega) e^{j\Omega t} d\Omega$$

Convolution Integral

The convolution integral expresses the output of an LTI system based on an input signal, $x(t)$, and the system's impulse response, $h(t)$. The **convolution integral** is expressed as

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

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

$$y(t) = x(t) * h(t)$$

Convolution is commutative. For more information on the characteristics of the convolution integral, read about the Properties of Convolution (Section 3.4).

Demonstration

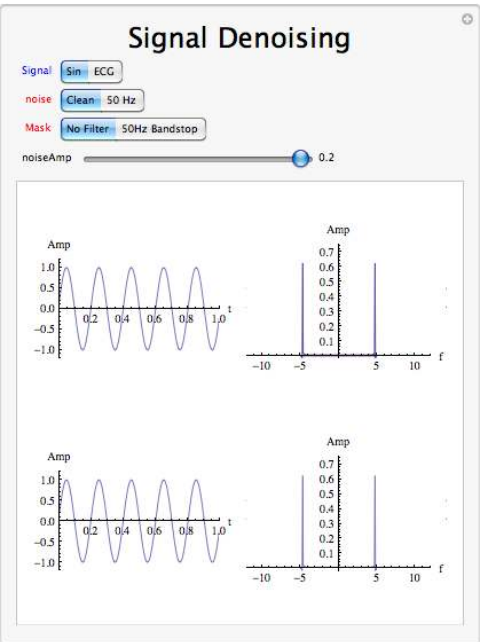


Figure 8.5.1: Interact (when online) with a Mathematica CDF demonstrating Use of the CTFT in signal denoising. To Download, right-click and save target as .cdf.

Convolution Theorem

Let f and g be two functions with convolution $f * g$. Let F be the Fourier transform operator. Then

$$F(f * g) = F(f) \cdot F(g)$$

$$F(f \cdot g) = \frac{1}{2\pi} F(f) * F(g)$$

By applying the inverse Fourier transform F^{-1}, we can write:

$$f * g = F^{-1}(F(f) \cdot F(g))$$

Conclusion

The Fourier transform of a convolution is the pointwise product of Fourier transforms. In other words, convolution in one domain (e.g., time domain) corresponds to point-wise multiplication in the other domain (e.g., frequency domain).

This page titled 8.5: Continuous Time Convolution and the CTFT is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

9: Discrete Time Fourier Transform (DTFT)

9.1: Discrete Time Aperiodic Signals

9.2: Discrete Time Fourier Transform (DTFT)

9.3: Common Discrete Time Fourier Transforms

9.4: Properties of the DTFT

9.5: Discrete Time Convolution and the DTFT

This page titled 9: Discrete Time Fourier Transform (DTFT) is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

9.1: Discrete Time Aperiodic Signals

Introduction

This module describes the type of signals acted on by the Discrete Time Fourier Transform.

Relevant Spaces

The Discrete Time Fourier Transform maps arbitrary discrete time signals in $l^2(\mathbb{Z})$ to finite-length, discrete-frequency signals in $L^2([0, 2\pi))$.

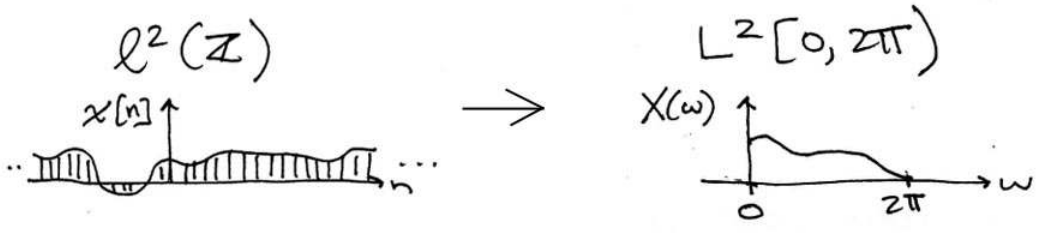


Figure 9.1.1: Mapping $l^2(\mathbb{Z})$ in the time domain to $L^2([0, 2\pi))$ in the frequency domain.

Periodic and Aperiodic Signals

When a function repeats itself exactly after some given period, or cycle, we say it's **periodic**. A **periodic function** can be mathematically defined as:

$$f[n] = f[n + mN] \quad m \in \mathbb{Z} \tag{9.1.1}$$

where $N > 0$ represents the **fundamental period** of the signal, which is the smallest positive value of N for the signal to repeat. Because of this, you may also see a signal referred to as an N-periodic signal. Any function that satisfies this equation is said to be **periodic** with period N. Periodic signals in discrete time repeats themselves in each cycle. However, only integers are allowed as time variable in discrete time. We denote signals in such case as $f[n]$, $n = ..., -2, -1, 0, 1, 2, ...$ Here's an example of a **discrete-time periodic signal** with period N:

discrete-time periodic signal

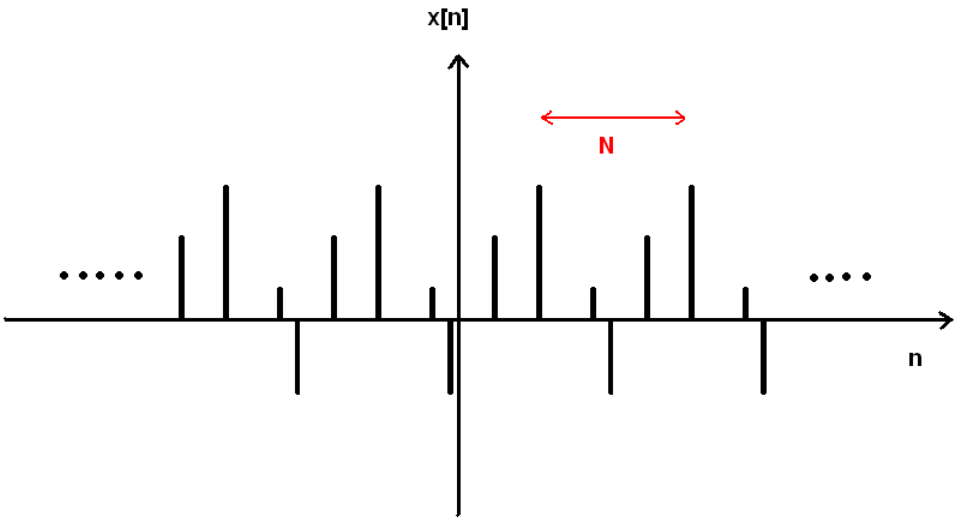


Figure 9.1.2: Notice the function is the same after a time shift of N

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

1. as functions on **all** of $\mathbb{R}$

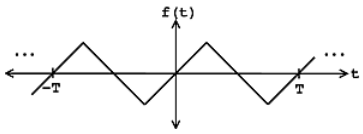


Figure 9.1.3: discrete time periodic function over all of $\mathbb{R}$ where $f[n_0] = f[n_0 + N]$

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

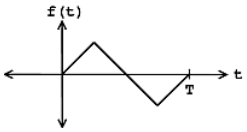


Figure 9.1.4: Remove the redundancy of the period function so that $f[n]$ is undefined outside $[0, N]$.

An **aperiodic** DT function, however, $f[n]$ does not repeat for **any** $N \in \mathbb{R}$; i.e. there exists no N such that Equation 9.1.1 holds. This broader class of signals can only be acted upon by the DTFT.

Suppose we have such an aperiodic function $f[n]$. We can construct a periodic extension of $f[n]$ called $f_{No}[n]$, where $f[n]$ is repeated every N_0 seconds. If we take the limit as $N_0 \to \infty$, we obtain a precise model of an aperiodic signal for which all rules that govern periodic signals can be applied, including Fourier Analysis (with an important modification). For more detail on this distinction, see the module on the **Discete Time Fourier Transform**.

Aperiodic Signal Demonstration

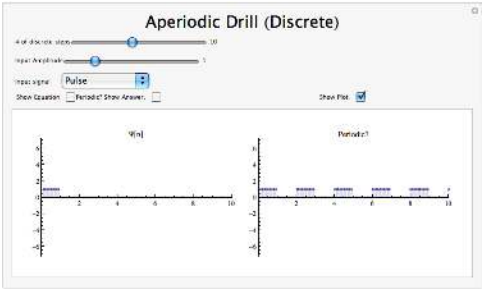


Figure 9.1.5: Click on the above thumbnail image (when online) to download an interactive Mathematica Player testing Periodic versus Aperiodic Signals. To download, right-click and save as .cdf.

Conclusion

A discrete periodic signal is completely defined by its values in one period, such as the interval [0,N]. Any aperiodic signal can be defined as an infinite sum of periodic functions, a useful definition that makes it possible to use Fourier Analysis on it by assuming all frequencies are present in the signal.

This page titled 9.1: Discrete Time Aperiodic Signals is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

9.2: Discrete Time Fourier Transform (DTFT)

Introduction

In this module, we will derive an expansion for arbitrary discrete-time functions, and in doing so, derive the **Discrete Time Fourier Transform** (DTFT).

Since complex exponentials (Section 1.8) are eigenfunctions of linear time-invariant (LTI) systems (Section 14.5), calculating the output of an LTI system $\mathcal{H}$ given $e^{j\omega n}$ as an input amounts to simple multiplication, where $\omega_0 = \frac{2\pi k}{N}$, and where $H[k] \in \mathrm{C}$ is the eigenvalue corresponding to k. As shown in the figure, a simple exponential input would yield the output

$$y[n] = H[k]e^{j\omega n}$$

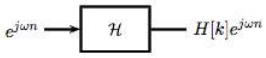


Figure 9.2.1: Simple LTI system.

Using this and the fact that $\mathcal{H}$ is linear, calculating $y[n]$ for combinations of **complex exponentials** is also straightforward.

$$c_1 e^{j\omega_1 n} + c_2 e^{j\omega_2 n} \rightarrow c_1 H\left[k_1\right] e^{j\omega_1 n} + c_2 H\left[k_2\right] e^{j\omega_1 n}$$
$$\sum_l c_l e^{j\omega_l n} \rightarrow \sum_l c_l H\left[k_l\right] e^{j\omega_l n}$$

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 $e^{j\omega_l n}$ by a different complex number $H\left[k_l\right] \in \mathbb{C}$. As such, if we can write a function $y[n]$ as a combination of complex exponentials it allows us to easily calculate the output of a system.

Now, we will look to use the power of complex exponentials to see how we may represent arbitrary signals in terms of a set of simpler functions by superposition of a number of complex exponentials. Below we will present the **Discrete-Time Fourier Transform** (DTFT). Because the DTFT deals with nonperiodic signals, we must find a way to include all real frequencies in the general equations. For the DTFT we simply utilize summation over all real numbers rather than summation over integers in order to express the aperiodic signals.

DTFT synthesis

It can be demonstrated that an arbitrary Discrete Time-periodic function $f[n]$ can be written as a linear combination of harmonic complex sinusoids

$$f[n] = \sum_{k=0}^{N-1} c_k e^{j\omega_0 kn} \tag{9.2.1}$$

where $\omega_0 = \frac{2\pi}{N}$ is the fundamental frequency. For almost all $f[n]$ of practical interest, there exists c_n to make Equation 9.2.1 true. If $f[n]$ is finite energy ($f[n] \in L^2[0, N]$), then the equality in Equation 9.2.1 holds in the sense of energy convergence; with discrete-time signals, there are no concerns for divergence as there are with continuous-time signals.

The c_n - called the Fourier coefficients - tell us "how much" of the sinusoid $e^{j\omega_0 kn}$ is in $f[n]$. The formula shows $f[n]$ as a sum of complex exponentials, each of which is easily processed by an LTI system (since it is an eigenfunction of **every** LTI system). Mathematically, it tells us that the set of complex exponentials $\{e^{j\omega_0 kn}, k \in \mathbb{Z}\}$ form a basis for the space of N-periodic discrete time functions.

Equations

Now, in order to take this useful tool and apply it to arbitrary non-periodic signals, we will have to delve deeper into the use of the superposition principle. Let $s_T(t)$ be a periodic signal having period T. We want to consider what happens to this signal's spectrum as the period goes to infinity. We denote the spectrum for any assumed value of the period by $c_n(T)$. We calculate the spectrum according to the Fourier formula for a periodic signal, known as the Fourier Series (for more on this derivation, see the section on **Fourier Series**.)

$$c_n = \frac{1}{T}\int_0^T s(t)\exp\left(-j\omega_0 t\right)dt$$

where $\omega_0 = \frac{2\pi}{T}$ and where we have used a symmetric placement of the integration interval about the origin for subsequent derivational convenience. We vary the frequency index n proportionally as we increase the period. Define

$$S_T(f) \equiv Tc_n = \frac{1}{T}\int_0^T S_T(f)\exp\left(j\omega_0 t\right)dt$$

making the corresponding Fourier Series

$$s_T(t) = \sum_{-\infty}^{\infty} f(t)\exp\left(j\omega_0 t\right)\frac{1}{T}$$

As the period increases, the spectral lines become closer together, becoming a continuum. Therefore,

$$\lim_{T\to\infty} s_T(t) \equiv s(t) = \int_{-\infty}^{\infty} S(f)\exp\left(j\omega_0 t\right)df$$

with

$$S(f) = \int_{-\infty}^{\infty} s(t)\exp\left(-j\omega_0 t\right)dt$$

Discrete-Time Fourier Transform

$$\mathcal{F}(\omega) = \sum_{n=-\infty}^{\infty} f[n]e^{-(j\omega n)}$$

Inverse DTFT

$$f[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{F}(\omega)e^{j\omega n}d\omega$$

Warning

It is not uncommon to see the above formula written slightly different. One of the most common differences is the way that the exponential is written. The above equations use the radial frequency variable ω in the exponential, where $\omega = 2\pi f$, but it is also common to include the more explicit expression, $j2\pi ft$, in the exponential. Sometimes DTFT notation is expressed as $F(e^{j\omega})$, to make it clear that it is not a CTFT (which is denoted as $F(\Omega)$. Click here for an overview of the notation used in Connexion's DSP modules.

DTFT Definition demonstration

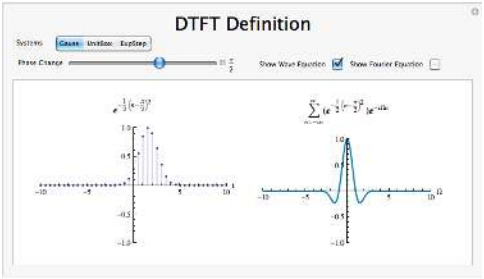


Figure 9.2.2: Click on the above thumbnail image (when online) to download an interactive Mathematica Player demonstrating Discrete Time Fourier Transform. To Download, right-click and save target as .cdf.

DTFT Summary

Because complex exponentials are eigenfunctions of LTI systems, it is often useful to represent signals using a set of complex exponentials as a basis. The discrete time Fourier transform synthesis formula expresses a discrete time, aperiodic function as the infinite sum of continuous frequency complex exponentials.

$$\mathcal{F}(\omega) = \sum_{n=-\infty}^{\infty} f[n]e^{-(j\omega n)}$$

The discrete time Fourier transform analysis formula takes the same discrete time domain signal and represents the signal in the continuous frequency domain.

$$f[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{F}(\omega)e^{j\omega n}d\omega$$

This page titled 9.2: Discrete Time Fourier Transform (DTFT) is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

9.3: Common Discrete Time Fourier Transforms

Common DTFTs

Table 9.3.1

Time Domain $x[n]$	Frequency Domain $X(w)$	Notes
$\delta[n]$	1	
$\delta[n-M]$	e^{-jwM}	integer M
$\sum_{m=-\infty}^{\infty}\delta[n-Mm]$	$\sum_{m=-\infty}^{\infty}e^{-jwMm}=\frac{1}{M}\sum_{k=-\infty}^{\infty}\delta\left(\frac{w}{2\pi}-\frac{k}{M}\right)$	integer M
e^{-jan}	$2\pi\delta(w+a)$	real number a
$u[n]$	$\frac{1}{1-e^{-jw}}+\sum_{k=-\infty}^{\infty}\pi\delta(w+2\pi k)$	
$a^n u(n)$	$\frac{1}{1-ae^{-jw}}$	if $\lvert a\rvert<1$
$\cos(an)$	$\pi[\delta(w-a)+\delta(w+a)]$	real number a
$W\cdot\operatorname{sinc}^2(Wn)$	$\operatorname{tri}\left(\frac{w}{2\pi W}\right)$	real number W, $0<W\leq 0.50$
$W\cdot\operatorname{sinc}[W(n+a)]$	$\operatorname{rect}\left(\frac{w}{2\pi W}\right)\cdot e^{jaw}$	real numbers W, a $0<W\leq 1$
$\operatorname{rect}\left[\frac{(n-M/2)}{M}\right]$	$\frac{\sin[w(M+1)/2]}{\sin(w/2)}e^{-jwM/2}$	integer M
$\frac{W}{(n+a)}\{\cos[\pi W(n+a)]-\operatorname{sinc}[W(n+a)]\}$	$jw\cdot\operatorname{rect}\left(\frac{w}{\pi W}\right)e^{j}aw$	real numbers W, a $0<W\leq 1$
$\frac{1}{\pi n^2}\left[(-1)^n-1\right]$	$\lvert w\rvert$	
$\begin{cases}0 & n=0\\ \frac{(-1)^n}{n} & \text{elsewhere}\end{cases}$	jw	differentiator filter
$\begin{cases}0 & n \text{ odd}\\ \frac{2}{\pi n} & n \text{ even}\end{cases}$	$\begin{cases}j & w<0\\ 0 & w=0\\ -j & w>0\end{cases}$	Hilbert Transform

Notes

rect(t) is the rectangle function for arbitrary real-valued t.

$$\text{rect(t)}=\begin{cases}0 & \text{if } |t|>1/2\\ 1/2 & \text{if } |t|=1/2\\ 1 & \text{if } |t|<1/2\end{cases}$$

tri(t) is the triangle function for arbitrary real-valued t.

$$\text{tri(t)}=\begin{cases}1+t & \text{if } -1\leq t\leq 0\\ 1-t & \text{if } 0<t\leq 1\\ 0 & \text{otherwise}\end{cases}$$

This page titled 9.3: Common Discrete Time Fourier Transforms is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

9.4: Properties of the DTFT

Introduction

This module will look at some of the basic properties of the Discrete-Time Fourier Transform (DTFT) (Section 9.2).

Note

We will be discussing these properties for aperiodic, discrete-time signals but understand that very similar properties hold for continuous-time signals and periodic signals as well.

Discussion of Fourier Transform Properties

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 of transforms (Section 8.3) to find the transform of a more complicated signal.

Example 9.4.1

We will begin with the following signal:

$$z[n] = af_1[n] + bf_2[n]$$

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.

$$Z(\omega) = aF_1(\omega) + bF_2(\omega)$$

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 DTFT and the inverse DTFT. The only difference is the scaling by 2π and a frequency reversal.

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 pulse 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.

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 is proven below:

Example 9.4.2

We will begin by letting $z[n] = f[n-\eta]$. Now let us take the Fourier transform with the previous expression substituted in for $z[n]$.

$$Z(\omega) = \int_{-\infty}^{\infty} f[n-\eta] e^{-(j\omega n)} \mathrm{d}n$$

Now let us make a simple change of variables, where $\sigma = n - \eta$. 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.

$$\begin{aligned} Z(\omega) &= \int_{-\infty}^{\infty} f[\sigma] e^{-(j\omega(\sigma+\eta)n)} d\eta \\ &= e^{-(j\omega\eta)} \int_{-\infty}^{\infty} f[\sigma] e^{-(j\omega\sigma)} d\sigma \\ &= e^{-(j\omega\eta)} F(\omega) \end{aligned} \tag{9.4.1}$$

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 discrete-time convolution (Section 4.3) module for a more in depth explanation and derivation.

$$\begin{aligned} y[n] &= (f_1[n], f_2[n]) \\ &= \sum_{\eta=-\infty}^{\infty} f_1[\eta] f_2[n-\eta] \end{aligned} \tag{9.4.2}$$

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 Z Transform (Section 11.1).

Parseval's Relation

$$\sum_{n=-\infty}^{\infty} (|f[n]|)^2 = \int_{-\pi}^{\pi} (|F(\omega)|)^2 d\omega$$

Parseval's relation tells us that the energy of a signal is equal to the energy of its Fourier transform.

$$f[n] \xleftrightarrow{\text{DTFT}} F(\omega)$$

Figure 9.4.1

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 9.4); however, here we would use the inverse Fourier transform in place of the Fourier transform. Since we went through the steps in the previous, time-shift proof, below we will just show the initial and final step to this proof:

$$z(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega - \phi) e^{j\omega t} d\omega$$

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:

$$z(t) = f(t)e^{j\phi t}$$

Properties Demonstration

An interactive example demonstration of the properties is included below:

Figure 9.4.2: Interactive Signal Processing Laboratory Virtual Instrument created using NI's Labview.

Summary Table of DTFT Properties

Table 9.4.1: Discrete-time Fourier transform properties and relations.

	Sequence Domain	Frequency Domain
Linearity	$a_1 s_1(n) + a_2 s_2(n)$	$a_1 S_1\left(e^{j2\pi f}\right) + a_2 S_2\left(e^{j2\pi f}\right)$
Conjugate Symmetry	$s(n)$ real	$S\left(e^{j2\pi f}\right) = S\left(e^{-(j2\pi f)}\right)^*$
Even Symmetry	$s(n) = s(-n)$	$S\left(e^{j2\pi f}\right) = S\left(e^{-(j2\pi f)}\right)$
Odd Symmetry	$s(n) = -s(-n)$	$S\left(e^{j2\pi f}\right) = -S\left(e^{-(j2\pi f)}\right)$
Time Delay	$s(n - n_0)$	$e^{-(j2\pi f n_0)} S\left(e^{j2\pi f}\right)$
Multiplication by n	$ns(n)$	$\frac{1}{-(2j\pi)} \frac{dS(e^{j2\pi f})}{df}$
Sum	$\sum_{n=-\infty}^{\infty} s(n)$	$S(e^{j2\pi 0})$
Value at Origin	$s(0)$	$\int_{-\frac{1}{2}}^{\frac{1}{2}} S\left(e^{j2\pi f}\right) df$
Parseval's Theorem	$\sum_{n=-\infty}^{\infty} \left(\lvert s(n)\rvert\right)^2$	$\int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\left\lvert S\left(e^{j2\pi f}\right)\right\rvert\right)^2 df$
Complex Modulation	$e^{j2\pi f_0 n} s(n)$	$S\left(e^{j2\pi(f-f_0)}\right)$
Amplitude Modulation	$s(n)\cos(2\pi f_0 n)$	$\frac{S\left(e^{j2\pi(f-f_0)}\right)+S\left(e^{j2\pi(f+f_0)}\right)}{2}$
	$s(n)\sin(2\pi f_0 n)$	$\frac{S\left(e^{j2\pi(f-f_0)}\right)-S\left(e^{j2\pi(f+f_0)}\right)}{2}$

This page titled 9.4: Properties of the DTFT is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

9.5: Discrete Time Convolution and the DTFT

Introduction

This module discusses convolution of discrete signals in the time and frequency domains.

The Discrete-Time Convolution

Discrete Time Fourier Transform

The DTFT transforms an infinite-length discrete signal in the time domain into an finite-length (or 2π-periodic) continuous signal in the frequency domain.

DTFT

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-(j\omega n)}$$

Inverse DTFT

$$x(n) = \frac{1}{2\pi}\int_{0}^{2\pi} X(\omega)e^{j\omega n}d\omega$$

Demonstration

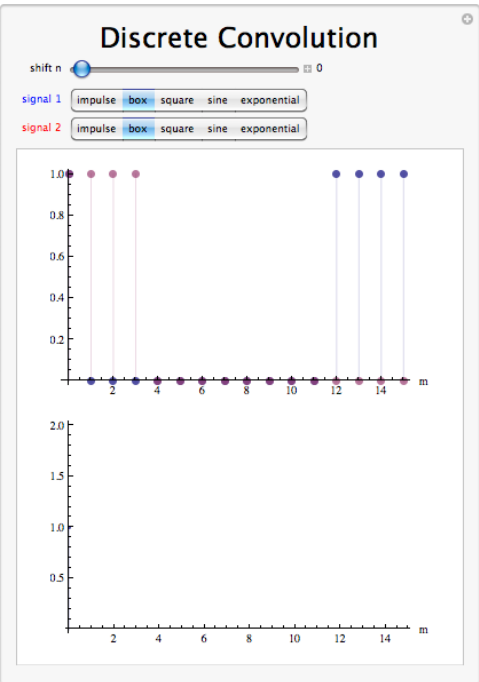


Figure 9.5.1: Interact (when online) with a Mathematica CDF demonstrating the Discrete Convolution. To Download, right-click and save as .cdf.

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 impulse response. The convolution sum is expressed as

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

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

$$y[n] = x[n] * h[n]$$

Convolution is commutative. For more information on the characteristics of convolution, read about the Properties of Convolution (Section 3.4).

Convolution Theorem

Let f and g be two functions with convolution $f * g$. Let F be the Fourier transform operator. Then

$$F(f * g) = F(f) \cdot F(g)$$

$$F(f \cdot g) = F(f) * F(g)$$

By applying the inverse Fourier transform F^{-1}, we can write:

$$f * g = F^{-1}(F(f) \cdot F(g))$$

Conclusion

The Fourier transform of a convolution is the pointwise product of Fourier transforms. In other words, convolution in one domain (e.g., time domain) corresponds to point-wise multiplication in the other domain (e.g., frequency domain).

This page titled 9.5: Discrete Time Convolution and the DTFT is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

10: Sampling and Reconstruction

10.1: Signal Sampling

10.2: Sampling Theorem

10.3: Signal Reconstruction

10.4: Perfect Reconstruction

10.5: Aliasing Phenomena

10.6: Anti-Aliasing Filters

10.7: Discrete Time Processing of Continuous Time Signals

This page titled 10: Sampling and Reconstruction is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

10.1: Signal Sampling

Introduction

Digital computers can process discrete time signals using extremely flexible and powerful algorithms. However, most signals of interest are continuous time signals, which is how data almost always appears in nature. This module introduces the concepts behind converting continuous time signals into discrete time signals through a process called sampling.

Sampling

Sampling a continuous time signal produces a discrete time signal by selecting the values of the continuous time signal at evenly spaced points in time. Thus, sampling a continuous time signal x with sampling period T_s gives the discrete time signal x_s defined by $x_s(n) = x(nT_s)$. The sampling angular frequency is then given by $\omega_s = 2\pi/T_s$.

It should be intuitively clear that multiple continuous time signals sampled at the same rate can produce the same discrete time signal since uncountably many continuous time functions could be constructed that connect the points on the graph of any discrete time function. Thus, sampling at a given rate does not result in an injective relationship. Hence, sampling is, in general, not invertible.

Example 10.1.1

For instance, consider the signals x,y defined by

$$x(t) = \frac{\sin(t)}{t}$$

$$y(t) = \frac{\sin(5t)}{t}$$

and their sampled versions x_S,y_s with sampling period $T_s = \pi/2$

$$x_s(n) = \frac{\sin(n\pi/2)}{n\pi/2}$$

$$y_s(n) = \frac{\sin(n5\pi/2)}{n\pi/2}.$$

Notice that since

$$\sin(n5\pi/2) = \sin(n2\pi + n\pi/2) = \sin(n\pi/2)$$

it follows that

$$y_s(n) = \frac{\sin(n\pi/2)}{n\pi/2} = x_s(n).$$

Hence, x and y provide an example of distinct functions with the same sampled versions at a specific sampling rate.

It is also useful to consider the relationship between the frequency domain representations of the continuous time function and its sampled versions. Consider a signal x sampled with sampling period T_s to produce the discrete time signal $x_s(n) = x(nT_s)$. The spectrum $X_s(\omega)$ for $\omega \in [-\pi, \pi)$ of x_s is given by

$$X_s(\omega) = \sum_{n=-\infty}^{\infty} x\left(nT_s\right) e^{-j\omega n}.$$

Using the continuous time Fourier transform, $x(tT_s)$ can be represented as

$$x\left(tT_s\right) = \frac{1}{2\pi T_s} \int_{-\infty}^{\infty} X\left(\frac{\omega_1}{T_s}\right) e^{j\omega_1 t} d\omega_1.$$

Thus, the unit sampling period version of $x(tT_s)$, which is $x(nT_s)$ can be represented as

$$x\left(nT_s\right) = \frac{1}{2\pi T_s}\int_{-\infty}^{\infty} X\left(\frac{\omega_1}{T_s}\right) e^{j\omega_1 n} d\omega_1.$$

This is algebraically equivalent to the representation

$$x\left(nT_s\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty}\frac{1}{2\pi}\int_{-\pi}^{\pi} X\left(\frac{\omega_1 - 2\pi k}{T_s}\right) e^{j(\omega_1 - 2\pi k)n} d\omega_1,$$

which reduces by periodicity of complex exponentials to

$$x\left(nT_s\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty}\frac{1}{2\pi}\int_{-\pi}^{\pi} X\left(\frac{\omega_1 - 2\pi k}{T_s}\right) e^{j\omega_1 n} d\omega_1.$$

Hence, it follows that

$$X_s(\omega) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\left(\int_{-\pi}^{\pi} X\left(\frac{\omega_1 - 2\pi k}{T_s}\right) e^{j\omega_1 n} d\omega_1\right) e^{-j\omega n}.$$

Noting that the above expression contains a Fourier series and inverse Fourier series pair, it follows that

$$X_s(\omega) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X\left(\frac{\omega - 2\pi k}{T_s}\right).$$

Hence, the spectrum of the sampled signal is, intuitively, the scaled sum of an infinite number of shifted and time scaled copies of original signal spectrum. Aliasing, which will be discussed in depth in later modules, occurs when these shifted spectrum copies overlap and sum together. Note that when the original signal x is bandlimited to $(-\pi/T_s, \pi/T_s)$ no overlap occurs, so each period of the sampled signal spectrum has the same form as the original signal spectrum. This suggest that if we sample a bandlimited signal at a sufficiently high sampling rate, we can recover it from its samples as will be further described in the modules on the Nyquist-Shannon sampling theorem and on perfect reconstruction.

Sampling Summary

Sampling a continuous time signal produces a discrete time signal by selecting the values of the continuous time signal at equally spaced points in time. However, we have shown that this relationship is not injective as multiple continuous time signals can be sampled at the same rate to produce the same discrete time signal. This is related to a phenomenon called aliasing which will be discussed in later modules. Consequently, the sampling process is not, in general, invertible. Nevertheless, as will be shown in the module concerning reconstruction, the continuous time signal can be recovered from its sampled version if some additional assumptions hold.

This page titled 10.1: Signal Sampling is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

10.2: Sampling Theorem

Introduction

With the introduction of the concept of signal sampling, which produces a discrete time signal by selecting the values of the continuous time signal at evenly spaced points in time, it is now possible to discuss one of the most important results in signal processing, the Nyquist-Shannon sampling theorem. Often simply called the sampling theorem, this theorem concerns signals, known as bandlimited signals, with spectra that are zero for all frequencies with absolute value greater than or equal to a certain level. The theorem implies that there is a sufficiently high sampling rate at which a bandlimited signal can be recovered exactly from its samples, which is an important step in the processing of continuous time signals using the tools of discrete time signal processing.

Nyquist-Shannon Sampling Theorem

Statement of the Sampling Theorem

The Nyquist-Shannon sampling theorem concerns signals with continuous time Fourier transforms that are only nonzero on the interval $(-B, B)$ for some constant B. Such a function is said to be bandlimited to $(-B, B)$. Essentially, the sampling theorem has already been implicitly introduced in the previous module concerning sampling. Given a continuous time signals x with continuous time Fourier transform X, recall that the spectrum X_s of sampled signal x_s with sampling period T_s is given by

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X\left(\frac{\omega - 2\pi k}{T_s}\right).$$

It had previously been noted that if x is bandlimited to $(-\pi/T_s, \pi/T_s)$, the period of X_s centered about the origin has the same form as X scaled in frequency since no aliasing occurs. This is illustrated in Figure 10.2.1. Hence, if any two $(-\pi/T_s, \pi/T_s)$ bandlimited continuous time signals sampled to the same signal, they would have the same continuous time Fourier transform and thus be identical. Thus, for each discrete time signal there is a unique $(-\pi/T_s, \pi/T_s)$ bandlimited continuous time signal that samples to the discrete time signal with sampling period T_s. Therefore, this $(-\pi/T_s, \pi/T_s)$ bandlimited signal can be found from the samples by inverting this bijection.

This is the essence of the sampling theorem. More formally, the sampling theorem states the following. If a signal x is bandlimited to $(-B, B)$, it is completely determined by its samples with sampling rate $\omega_s = 2B$. That is to say, x can be reconstructed exactly from its samples x_s with sampling rate $\omega_s = 2B$. The angular frequency $2B$ is often called the angular Nyquist rate. Equivalently, this can be stated in terms of the sampling period $T_s = 2\pi/\omega_s$. If a signal x is bandlimited to $(-B, B)$, it is completely determined by its samples with sampling period $T_s = \pi/B$. That is to say, x can be reconstructed exactly from its samples x_s with sampling period T_s.

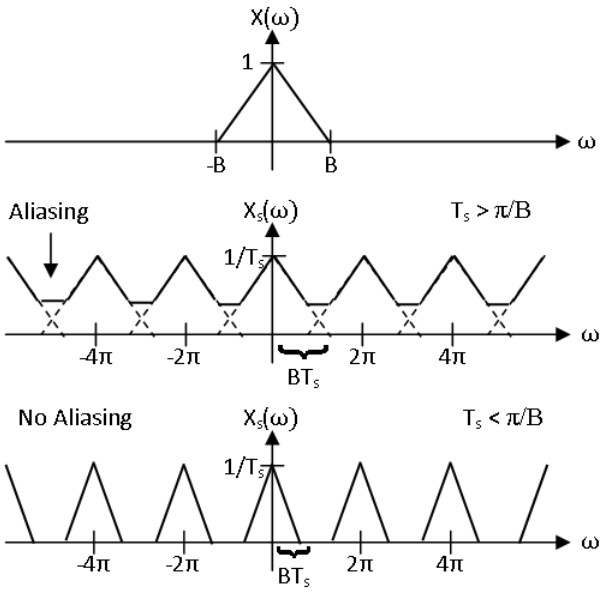


Figure 10.2.1: The spectrum of a bandlimited signals is shown as well as the spectra of its samples at rates above and below the Nyquist frequency. As is shown, no aliasing occurs above the Nyquist frequency, and the period of the samples spectrum centered about the origin has the same form as the spectrum of the original signal scaled in frequency. Below the Nyquist frequency, aliasing can occur and causes the spectrum to take a different than the original spectrum.

Proof of the Sampling Theorem

The above discussion has already shown the sampling theorem in an informal and intuitive way that could easily be refined into a formal proof. However, the original proof of the sampling theorem, which will be given here, provides the interesting observation that the samples of a signal with period T_s provide Fourier series coefficients for the original signal spectrum on $(-\pi/T_s, \pi/T_s)$.

Let x be a $(-\pi/T_s, \pi/T_s)$ bandlimited signal and x_s be its samples with sampling period T_s. We can represent x in terms of its spectrum X using the inverse continuous time Fourier transform and the fact that x is bandlimited. The result is

$$x(t) = \frac{1}{2\pi} \int_{-\pi/T_x}^{\pi/T_s} X(\omega) e^{j\omega t} d\omega$$

This representation of x may then be sampled with sampling period T_s to produce

$$x_s(n) = x_s(nT_s) = \frac{1}{2\pi} \int_{-\pi/T_s}^{\pi/T_s} X(\omega) e^{j\omega n T_s} d\omega$$

Noticing that this indicates that $x_s(n)$ is the nth continuous time Fourier series coefficient for $X(\omega)$ on the interval $(-\pi/T_s, \pi/T_s)$, it is shown that the samples determine the original spectrum $X(\omega)$ and, by extension, the original signal itself.

Perfect Reconstruction

Another way to show the sampling theorem is to derive the reconstruction formula that gives the original signal $\tilde{x} = x$ from its samples x_s with sampling period T_s, provided x is bandlimited to $(-\pi/T_s, \pi/T_s)$. This is done in the module on perfect reconstruction. However, the result, known as the Whittaker-Shannon reconstruction formula, will be stated here. If the requisite conditions hold, then the perfect reconstruction is given by

$$x(t) = \sum_{n=-\infty}^{\infty} x_s(n) \operatorname{sinc}(t/T_s - n)$$

where the sinc function is defined as

$$\operatorname{sinc}(t) = \frac{\sin(\pi t)}{\pi t}.$$

From this, it is clear that the set

$$\{\text{sinc}\,(t/T_s - n) \mid n \in \mathbb{Z}\}$$

forms an orthogonal basis for the set of $(-\pi/T_s, \pi/T_s)$ bandlimited signals, where the coefficients of a $(-\pi/T_s, \pi/T_s)$ signal in this basis are its samples with sampling period T_s.

Practical Implications

Discrete Time Processing of Continuous Time Signals

The Nyquist-Shannon Sampling Theorem and the Whittaker-Shannon Reconstruction formula enable discrete time processing of continuous time signals. Because any linear time invariant filter performs a multiplication in the frequency domain, the result of applying a linear time invariant filter to a bandlimited signal is an output signal with the same bandlimit. Since sampling a bandlimited continuous time signal above the Nyquist rate produces a discrete time signal with a spectrum of the same form as the original spectrum, a discrete time filter could modify the samples spectrum and perfectly reconstruct the output to produce the same result as a continuous time filter. This allows the use of digital computing power and flexibility to be leveraged in continuous time signal processing as well. This is more thoroughly described in the final module of this chapter.

Psychoacoustics

The properties of human physiology and psychology often inform design choices in technologies meant for interaction with people. For instance, digital devices dealing with sound use sampling rates related to the frequency range of human vocalizations and the frequency range of human auditory sensitivity. Because most of the sounds in human speech concentrate most of their signal energy between 5 Hz and 4 kHz, most telephone systems discard frequencies above 4 kHz and sample at a rate of 8 kHz. Discarding the frequencies greater than or equal to 4 kHz through use of an anti-aliasing filter is important to avoid aliasing, which would negatively impact the quality of the output sound as is described in a later module. Similarly, human hearing is sensitive to frequencies between 20 Hz and 20 kHz. Therefore, sampling rates for general audio waveforms placed on CDs were chosen to be greater than 40 kHz, and all frequency content greater than or equal to some level is discarded. The particular value that was chosen, 44.1 kHz, was selected for other reasons, but the sampling theorem and the range of human hearing provided a lower bound for the range of choices.

Sampling Theorem Summary

The Nyquist-Shannon Sampling Theorem states that a signal bandlimited to $(-\pi/T_s, \pi/T_s)$ can be reconstructed exactly from its samples with sampling period T_s. The Whittaker-Shannon interpolation formula, which will be further described in the section on perfect reconstruction, provides the reconstruction of the unique $(-\pi/T_s, \pi/T_s)$ bandlimited continuous time signal that samples to a given discrete time signal with sampling period T_s. This enables discrete time processing of continuous time signals, which has many powerful applications.

This page titled 10.2: Sampling Theorem is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

10.3: Signal Reconstruction

Introduction

The sampling process produces a discrete time signal from a continuous time signal by examining the value of the continuous time signal at equally spaced points in time. Reconstruction, also known as interpolation, attempts to perform an opposite process that produces a continuous time signal coinciding with the points of the discrete time signal. Because the sampling process for general sets of signals is not invertible, there are numerous possible reconstructions from a given discrete time signal, each of which would sample to that signal at the appropriate sampling rate. This module will introduce some of these reconstruction schemes.

Reconstruction

Reconstruction Process

The process of reconstruction, also commonly known as interpolation, produces a continuous time signal that would sample to a given discrete time signal at a specific sampling rate. Reconstruction can be mathematically understood by first generating a continuous time impulse train

$$x_{imp}(t) = \sum_{n=-\infty}^{\infty} x_s(n)\delta\left(t - nT_s\right)$$

from the sampled signal x_s with sampling period T_s and then applying a lowpass filter G that satisfies certain conditions to produce an output signal $\tilde{x}$. If G has impulse response g, then the result of the reconstruction process, illustrated in Figure 10.3.1, is given by the following computation, the final equation of which is used to perform reconstruction in practice.

$$\begin{aligned}\tilde{x}(t) &= \left(x_{imp} * g\right)(t) \\ &= \int_{-\infty}^{\infty} x_{imp}(\tau)g(t-\tau)d\tau \\ &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_s(n)\delta\left(\tau - nT_s\right) g(t-\tau)d\tau \\ &= \sum_{n=-\infty}^{\infty} x_s(n) \int_{-\infty}^{\infty} \delta\left(\tau - nT_s\right) g(t-\tau)d\tau \\ &= \sum_{n=-\infty}^{\infty} x_s(n)g\left(t - nT_s\right) \end{aligned} \tag{10.3.1}$$

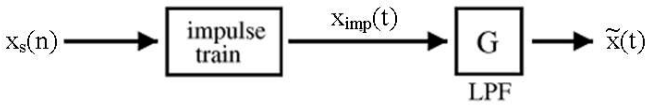


Figure 10.3.1: Block diagram of reconstruction process for a given lowpass filter G.

Reconstruction Filters

In order to guarantee that the reconstructed signal $\tilde{x}$ samples to the discrete time signal x_s from which it was reconstructed using the sampling period T_s, the lowpass filter G must satisfy certain conditions. These can be expressed well in the time domain in terms of a condition on the impulse response g of the lowpass filter G. The sufficient condition to be a reconstruction filters that we will require is that, for all $n \in \mathbb{Z}$,

$$g\left(nT_s\right) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} = \delta(n)\ .$$

This means that gg sampled at a rate T_s produces a discrete time unit impulse signal. Therefore, it follows that sampling $\tilde{x}$ with sampling period T_s results in

$$\begin{aligned}\tilde{x}(nT_s) &= \sum_{m=-\infty}^{\infty} x_s(m)g(nT_s - mT_s) \\ &= \sum_{m=-\infty}^{\infty} x_s(m)g((n-m)T_s) \\ &= \sum_{m=-\infty}^{\infty} x_s(m)\delta(n-m) \\ &= x_s(n), \end{aligned} \tag{10.3.2}$$

which is the desired result for reconstruction filters.

Cardinal Basis Splines

Since there are many continuous time signals that sample to a given discrete time signal, additional constraints are required in order to identify a particular one of these. For instance, we might require our reconstruction to yield a spline of a certain degree, which is a signal described in piecewise parts by polynomials not exceeding that degree. Additionally, we might want to guarantee that the function and a certain number of its derivatives are continuous.

This may be accomplished by restricting the result to the span of sets of certain splines, called **basis splines** or **B-splines**. Specifically, if a nth degree spline with continuous derivatives up to at least order $n-1$ is required, then the desired function for a given T_s belongs to the span of $\{B_n(t/T_s - k) \mid k \in \mathbb{Z}\}$ where

$$B_n = B_0 * B_{n-1}$$

for $n \geq 1$ and

$$B_0(t) = \begin{cases} 1 & -1/2 < t < 1/2 \\ 0 & \text{otherwise} \end{cases}.$$

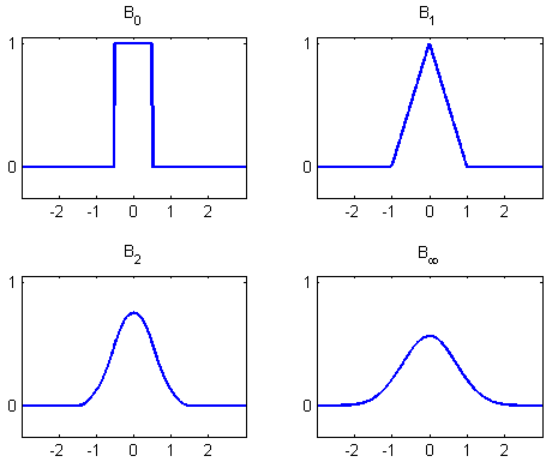


Figure 10.3.2: The basis splines B_n are shown in the above plots. Note that, except for the order 0 and order 1 functions, these functions do not satisfy the conditions to be reconstruction filters. Also notice that as the order increases, the functions approach the Gaussian function, which is exactly B_∞.

However, the basis splines B_n do not satisfy the conditions to be a reconstruction filter for $n \geq 2$ as is shown in Figure 10.3.2. Still, the B_n are useful in defining the cardinal basis splines, which do satisfy the conditions to be reconstruction filters. If we let b_n be the samples of B_n on the integers, it turns out that b_n has an inverse b_n^{-1} with respect to the operation of convolution for each n. This is to say that $b_n^{-1} * b_n = \delta$. The cardinal basis spline of order nn for reconstruction with sampling period T_s is defined as

$$\eta_n(t) = \sum_{k=-\infty}^{\infty} b_n^{-1}(k) B_n(t/T_s - k).$$

In order to confirm that this satisfies the condition to be a reconstruction filter, note that

$$\eta_n(mT_s) = \sum_{k=-\infty}^{\infty} b_n^{-1}(k) B_n(m-k) = \left(b_n^{-1} * b_n\right)(m) = \delta(m).$$

Thus, η_n is a valid reconstruction filter. Since η_n is an nth degree spline with continuous derivatives up to order $n-1$, the result of the reconstruction will be a nth degree spline with continuous derivatives up to order $n-1$.

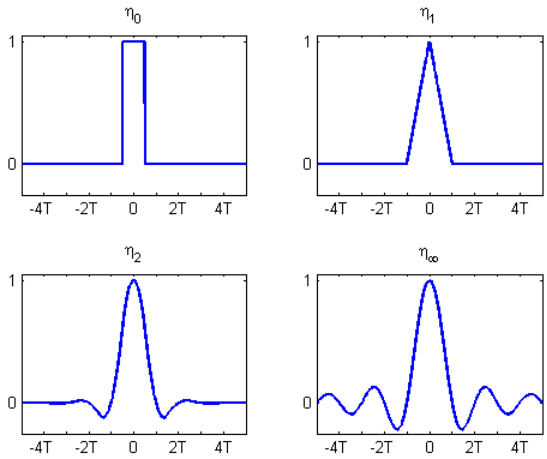


Figure 10.3.3: The above plots show cardinal basis spline functions η_0, η_1, η_2, and η_∞. Note that the functions satisfy the conditions to be reconstruction filters. Also, notice that as the order increases, the cardinal basis splines approximate the sinc function, which is exactly η_∞. Additionally, these filters are acausal.

The lowpass filter with impulse response equal to the cardinal basis spline η_0 of order 0 is one of the simplest examples of a reconstruction filter. It simply extends the value of the discrete time signal for half the sampling period to each side of every sample, producing a piecewise constant reconstruction. Thus, the result is discontinuous for all nonconstant discrete time signals.

Likewise, the lowpass filter with impulse response equal to the cardinal basis spline η_1 of order 1 is another of the simplest examples of a reconstruction filter. It simply joins the adjacent samples with a straight line, producing a piecewise linear reconstruction. In this way, the reconstruction is continuous for all possible discrete time signals. However, unless the samples are collinear, the result has discontinuous first derivatives.

In general, similar statements can be made for lowpass filters with impulse responses equal to cardinal basis splines of any order. Using the nth order cardinal basis spline η_n, the result is a piecewise degree nn polynomial. Furthermore, it has continuous derivatives up to at least order $n-1$. However, unless all samples are points on a polynomial of degree at most n, the derivative of order nn will be discontinuous.

Reconstructions of the discrete time signal given in Figure 10.3.4 using several of these filters are shown in Figure 10.3.5. As the order of the cardinal basis spline increases, notice that the reconstruction approaches that of the infinite order cardinal spline η_∞, the sinc function. As will be shown in the subsequent section on perfect reconstruction, the filters with impulse response equal to the sinc function play an especially important role in signal processing.

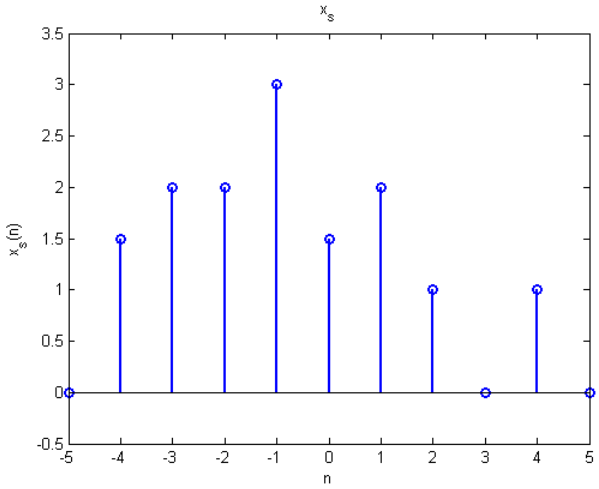


Figure 10.3.4: The above plot shows an example discrete time function. This discrete time function will be reconstructed using sampling period T_s using several cardinal basis splines in Figure 10.3.5.

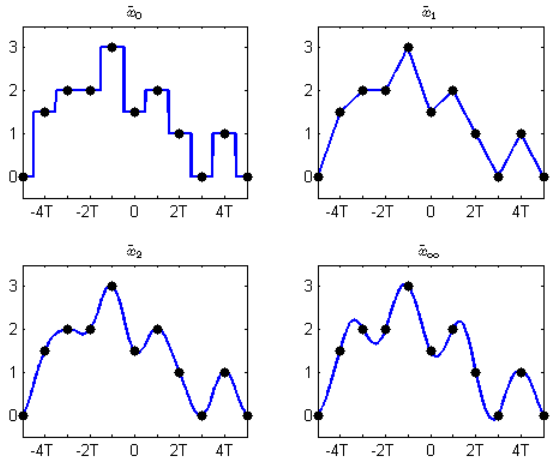


Figure 10.3.5: The above plots show interpolations of the discrete time signal given in Figure 10.3.4 using lowpass filters with impulse responses given by the cardinal basis splines shown in Figure 10.3.3. Notice that the interpolations become increasingly smooth and approach the sinc interpolation as the order increases.

Reconstruction Summary

Reconstruction of a continuous time signal from a discrete time signal can be accomplished through several schemes. However, it is important to note that reconstruction is not the inverse of sampling and only produces one possible continuous time signal that samples to a given discrete time signal. As is covered in the subsequent module, perfect reconstruction of a bandlimited continuous time signal from its sampled version is possible using the Whittaker-Shannon reconstruction formula, which makes use of the ideal lowpass filter and its sinc function impulse response, if the sampling rate is sufficiently high.

This page titled 10.3: Signal Reconstruction is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

10.4: Perfect Reconstruction

Introduction

If certain additional assumptions about the original signal and sampling rate hold, then the original signal can be recovered exactly from its samples using a particularly important type of filter. More specifically, it will be shown that if a bandlimited signal is sampled at a rate greater than twice its bandlimit, the Whittaker-Shannon reconstruction formula perfectly reconstructs the original signal. This formula makes use of the ideal lowpass filter, which is related to the sinc function. This is extremely useful, as sampled versions of continuous time signals can be filtered using discrete time signal processing, often in a computer. The results may then be reconstructed to produce the same continuous time output as some desired continuous time system.

Perfect Reconstruction

In order to understand the conditions for perfect reconstruction and the filter it employs, consider the following. As a beginning, a sufficient condition under which perfect reconstruction is possible will be discussed. Subsequently, the filter and process used for perfect reconstruction will be detailed.

Recall that the sampled version x_s of a continuous time signal x with sampling period T_s has a spectrum given by

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X\left(\frac{\omega - 2\pi k}{T_s}\right).$$

As before, note that if x is bandlimited to $(-\pi/T_s, \pi/T_s)$, meaning that X is only nonzero on $(-\pi/T_s, \pi/T_s)$, then each period of X_s has the same form as X. Thus, we can identify the original spectrum X from the spectrum of the samples X_s and, by extension, the original signal x from its samples x_s at rate T_s if x is bandlimited to $(-\pi/T_s, \pi/T_s)$.

If a signal x is bandlimited to $(-B, B)$, then it is also bandlimited to $(-\pi/T_s, \pi/T_s)$ provided that $T_s < \pi/B$. Thus, if we ensure that x is sampled to x_s with sufficiently high sampling angular frequency $\omega_s = 2\pi/T_s > 2B$ and have a way of identifying the unique $(-\pi/T_s, \pi/T_s)$ bandlimited signal corresponding to a discrete time signal at sampling period T_s, then x_s can be used to reconstruct $\tilde{x} = x$ exactly. The frequency $2B$ is known as the angular Nyquist rate. Therefore, the condition that the sampling rate $\omega_s = 2\pi/T_s > 2B$ be greater than the Nyquist rate is a sufficient condition for perfect reconstruction to be possible.

The correct filter must also be known in order to perform perfect reconstruction. The ideal lowpass filter defined by $G(\omega) = T_s(u(\omega + \pi/T_s) - u(\omega - \pi/T_s))$, which is shown in Figure 10.4.1, removes all signal content not in the frequency range $(-\pi/T_s, \pi/T_s)$. Therefore, application of this filter to the impulse train $\sum_{n=-\infty}^{\infty} x_s(n)\delta(t - nT_s)$ results in an output bandlimited to $(-\pi/T_s, \pi/T_s)$.

We now only need to confirm that the impulse response g of the filter G satisfies our sufficient condition to be a reconstruction filter. The inverse Fourier transform of $G(\omega)$ is

$$g(t) = \operatorname{sinc}(t/T_s) = \begin{cases} 1 & t = 0 \\ \frac{\sin(\pi t/T_s)}{\pi t/T_s} & t \neq 0 \end{cases},$$

which is shown in Figure 10.4.6. Hence,

$$g(nT_s) = \operatorname{sinc}(n) = \begin{cases} 1 & n = 0 \\ \frac{\sin(\pi n)}{\pi n} & n \neq 0 \end{cases} = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} = \delta(n).$$

Therefore, the ideal lowpass filter G is a valid reconstruction filter. Since it is a valid reconstruction filter and always produces an output that is bandlimited to $(-\pi/T_s, \pi/T_s)$, this filter always produces the unique $(-\pi/T_s, \pi/T_s)$ bandlimited signal that samples to a given discrete time sequence at sampling period T_s when the impulse train $\sum_{n=-\infty}^{\infty} x_s(n)\delta(t - nT_s)$ is input.

Therefore, we can always reconstruct any $(-\pi/T_s, \pi/T_s)$ bandlimited signal from its samples at sampling period T_s by the formula

$$x(t) = \sum_{n=-\infty}^{\infty} x_s(n) \operatorname{sinc}(t/T_s - n).$$

This perfect reconstruction formula is known as the **Whittaker-Shannon interpolation formula** and is sometimes also called the **cardinal series**. In fact, the sinc function is the infinite order cardinal basis spline η_∞. Consequently, the set $\{\operatorname{sinc}(t/T_s - n) \mid n \in \mathbb{Z}\}$

forms a basis for the vector space of $(-\pi/T_s, \pi/T_s)$ bandlimited signals where the signal samples provide the corresponding coefficients. It is a simple exercise to show that this basis is, in fact, an orthogonal basis.

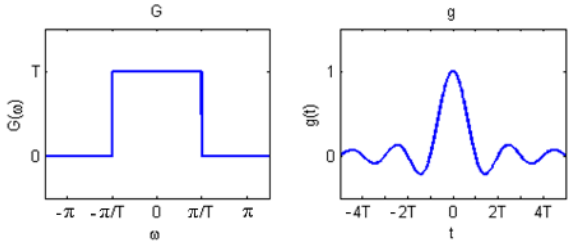


Figure 10.4.1: The above plots show the ideal lowpass filter and its inverse Fourier transform, the sinc function.

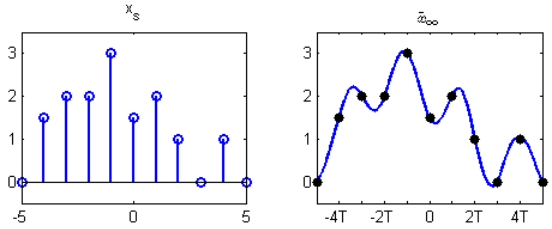


Figure 10.4.2: The plots show an example discrete time signal and its Whittaker-Shannon sinc reconstruction.

Perfect Reconstruction Summary

This module has shown that bandlimited continuous time signals can be reconstructed exactly from their samples provided that the sampling rate exceeds the Nyquist rate, which is twice the bandlimit. The Whittaker-Shannon reconstruction formula computes this perfect reconstruction using an ideal lowpass filter, with the resulting signal being a sum of shifted sinc functions that are scaled by the sample values. Sampling below the Nyquist rate can lead to aliasing which makes the original signal irrecoverable as is described in the subsequent module. The ability to perfectly reconstruct bandlimited signals has important practical implications for the processing of continuous time signals using the tools of discrete time signal processing.

This page titled 10.4: Perfect Reconstruction is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

10.5: Aliasing Phenomena

Introduction

Through discussion of the Nyquist-Shannon sampling theorem and Whittaker-Shannon reconstruction formula, it has already been shown that a $(-B, B)$ continuous time signal can be reconstructed from its samples at rate $\omega_s = 2\pi/T_s$ via the sinc interpolation filter if $\omega_s > 2B$. Now, this module will investigate a problematic phenomenon, called aliasing, that can occur if this sufficient condition for perfect reconstruction does not hold. When aliasing occurs the spectrum of the samples has different form than the original signal spectrum, so the samples cannot be used to reconstruct the original signal through Whittaker-Shannon interpolation.

Aliasing

Aliasing occurs when each period of the spectrum of the samples does not have the same form as the spectrum of the original signal. Given a continuous time signals x with continuous time Fourier transform X, recall that the spectrum X_s of sampled signal x_s with sampling period T_s is given by

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X\left(\frac{\omega - 2\pi k}{T_s}\right).$$

As has already been mentioned several times, if x is bandlimited to $(-\pi/T_s, \pi/T_s)$ then each period of X_s has the same form as X. However, if x is not bandlimited to $(-\pi/T_s, \pi/T_s)$, then the $X\left(\frac{\omega-2\pi k}{T_s}\right)$ can overlap and sum together. This is illustrated in Figure 10.5.1 in which sampling above the Nyquist frequency produces a samples spectrum of the same shape as the original signal, but sampling below the Nyquist frequency produces a samples spectrum with very different shape. Whittaker-Shannon interpolation of each of these sequences produces different results. The low frequencies not affected by the overlap are the same, but there is noise content in the higher frequencies caused by aliasing. Higher frequency energy masquerades as low energy content, a highly undesirable effect.

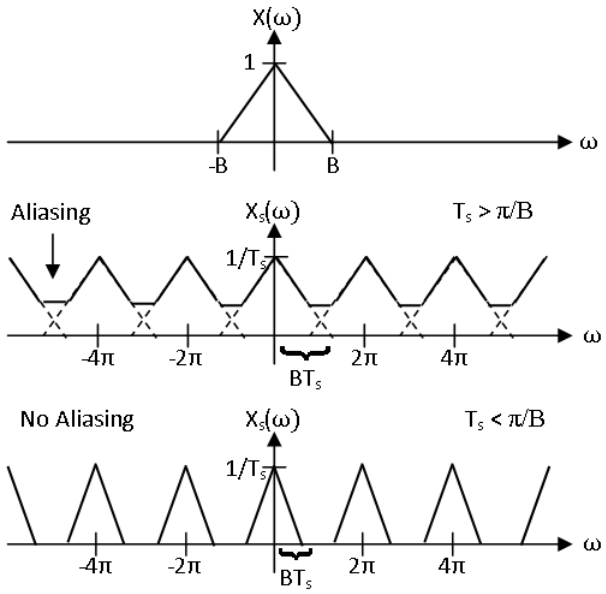


Figure 10.5.1: The spectrum of a bandlimited signals is shown as well as the spectra of its samples at rates above and below the Nyquist frequency. As is shown, no aliasing occurs above the Nyquist frequency, and the period of the samples spectrum centered about the origin has the same form as the spectrum of the original signal scaled in frequency. Below the Nyquist frequency, aliasing can occur and causes the spectrum to take a different than the original spectrum.

Unlike when sampling above the Nyquist frequency, sampling below the Nyquist frequency does not yield an injective (one-to-one) function from the $(-B, B)$ bandlimited continuous time signals to the discrete time signals. Any signal x with spectrum X which overlaps and sums to X_s samples to x_s. It should be intuitively clear that there are very many $(-B, B)$ bandlimited signals that sample to a given discrete time signal below the Nyquist frequency, as is demonstrated in Figure 10.5.2. It is quite easy to construct uncountably infinite families of such signals.

Aliasing obtains it name from the fact that multiple, in fact infinitely many, $(-B, B)$ bandlimited signals sample to the same discrete sequence if $\omega_s < 2B$. Thus, information about the original signal is lost in this noninvertible process, and these different signals effectively assume the same identity, an “alias”. Hence, under these conditions the Whittaker-Shannon interpolation formula will not produce a perfect reconstruction of the original signal but will instead give the unique $(-\omega_s/2, \omega_s/2)$ bandlimited signal that samples to the discrete sequence.

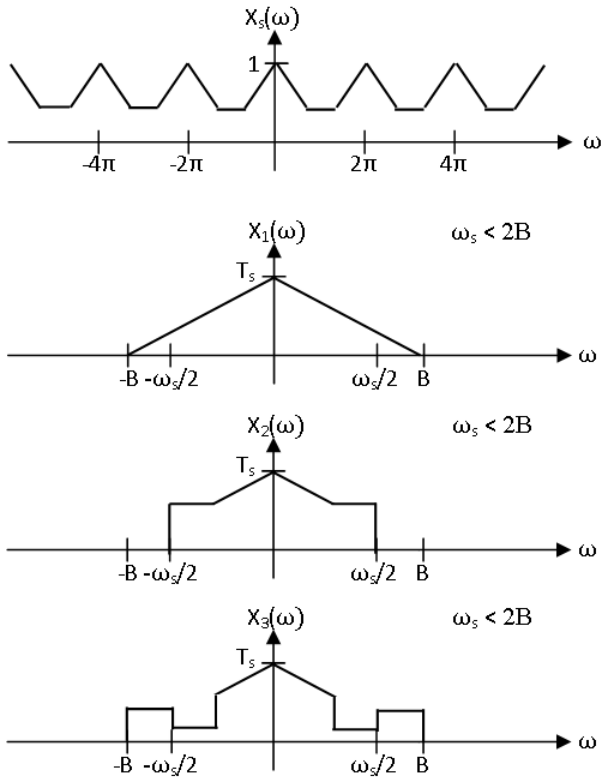


Figure 10.5.2: The spectrum of a discrete time signal x_s, taken from Figure 10.5.1, is shown along with the spectra of three $(-B, B)$ signals that sample to it at rate $\omega_s < 2B$. From the sampled signal alone, it is impossible to tell which, if any, of these was sampled at rate ω_s to produce x_s. In fact, there are infinitely many $(-B, B)$ bandlimited signals that sample to x_s at a sampling rate below the Nyquist rate.

Aliasing Demonstration

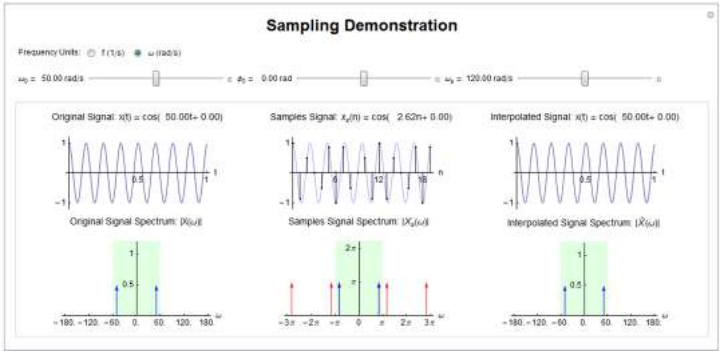


Figure 10.5.3: Interact (when online) with a Mathematica CDF demonstrating sampling and aliasing for a sinusoid. To Download, right-click and save target as .cdf.

Aliasing Summary

Aliasing, essentially the signal processing version of identity theft, occurs when each period of the spectrum of the samples does not have the same form as the spectrum of the original signal. As has been shown, there can be infinitely many $(-B, B)$ bandlimited signals that sample to a given discrete time signal x_s at a rate $\omega_s = 2\pi/T_s < 2B$ below the Nyquist frequency. However, there is a unique $(-B, B)$ bandlimited signal that samples to x_s, which is given by the Whittaker-Shannon interpolation of x_s, at rate $\omega_s \geq 2B$ as no aliasing occurs above the Nyquist frequency. Unfortunately, sufficiently high sampling rates cannot always be produced. Aliasing is detrimental to many signal processing applications, so in order to process continuous time signals using discrete time

tools, it is often necessary to find ways to avoid it other than increasing the sampling rate. Thus, anti-aliasing filters, are of practical importance.

This page titled 10.5: Aliasing Phenomena is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

10.6: Anti-Aliasing Filters

Introduction

It has been shown that a $(-B, B)$ bandlimited signal can be perfectly reconstructed from its samples at a rate $\omega_s = 2\pi/T_s \geq B$. However, it is not always practically possible to produce sufficiently high sampling rates or to ensure that the input is bandlimited in real situations. Aliasing, which manifests itself as a difference in shape between the periods of the samples signal spectrum and the original spectrum, would occur without any further measures to correct this. Thus, it often becomes necessary to filter out signal energy at frequencies above $\omega_s/2$ in order to avoid the detrimental effects of aliasing. This is the role of the anti-aliasing filter, a lowpass filter applied before sampling to ensure that the signal is $(-\omega_s/2, \omega_s/2)$ bandlimited or at least nearly so.

Anti-Aliasing Filters

Aliasing can occur when a signal with energy at frequencies other that $(-B, B)$ is sampled at rate $\omega_s < 2B$. Thus, when sampling below the Nyquist frequency, it is desirable to remove as much signal energy outside the frequency range $(-B, B)$ as possible while keeping as much signal energy in the frequency range $(-B, B)$ as possible. This suggests that the ideal lowpass filter with cutoff frequency $\omega_s/2$ would be the optimal anti-aliasing filter to apply before sampling. While this is true, the ideal lowpass filter can only be approximated in real situations.

In order to demonstrate the importance of anti-aliasing filters, consider the calculation of the error energy between the original signal and its Whittaker-Shannon reconstruction from its samples taken with and without the use of an anti-aliasing filter. Let x be the original signal and $y = Gx$ be the anti-alias filtered signal where G is the ideal lowpass filter with cutoff frequency $\omega_s/2$. It is easy to show that the reconstructed spectrum using no anti-aliasing filter is given by

$$\widetilde{X}(\omega) = \begin{cases} T_s X_s\left(T_s \omega\right) & |\omega| < \omega_s/2 \\ 0 & \text{otherwise} \end{cases} = \begin{cases} \sum_{k=-\infty}^{\infty} X\left(\omega - k\omega_s\right) & |\omega| < \omega_s/2 \\ 0 & \text{otherwise} \end{cases} .$$

Thus, the reconstruction error spectrum for this case is

$$(X - \widetilde{X})(\omega) = \begin{cases} -\sum_{k=1}^{\infty} \left(X\left(\omega + k\omega_s\right) + X\left(\omega - k\omega_s\right)\right) & |\omega| < \omega_s/2 \\ X(\omega) & \text{otherwise} \end{cases} .$$

Similarly, the reconstructed spectrum using the ideal lowpass anti-aliasing filter is given by

$$\widetilde{Y}(\omega) = Y(\omega) = \begin{cases} X(\omega) & |\omega| < \omega_s/2 \\ 0 & \text{otherwise} \end{cases} .$$

Thus, the reconstruction error spectrum for this case is

$$(X - \widetilde{Y})(\omega) = \begin{cases} 0 & |\omega| < \omega_s/2 \\ X(\omega) & \text{otherwise} \end{cases} .$$

Hence, by Parseval's theorem, it follows that $\|x - \tilde{y}\| \leq \|x - \tilde{x}\|$. Also note that the spectrum of $\widetilde{Y}$ is identical to that of the original signal X at frequencies $\omega \in (-\omega_s/2, \omega_s/2)$. This is graphically shown in Figure 10.6.1.

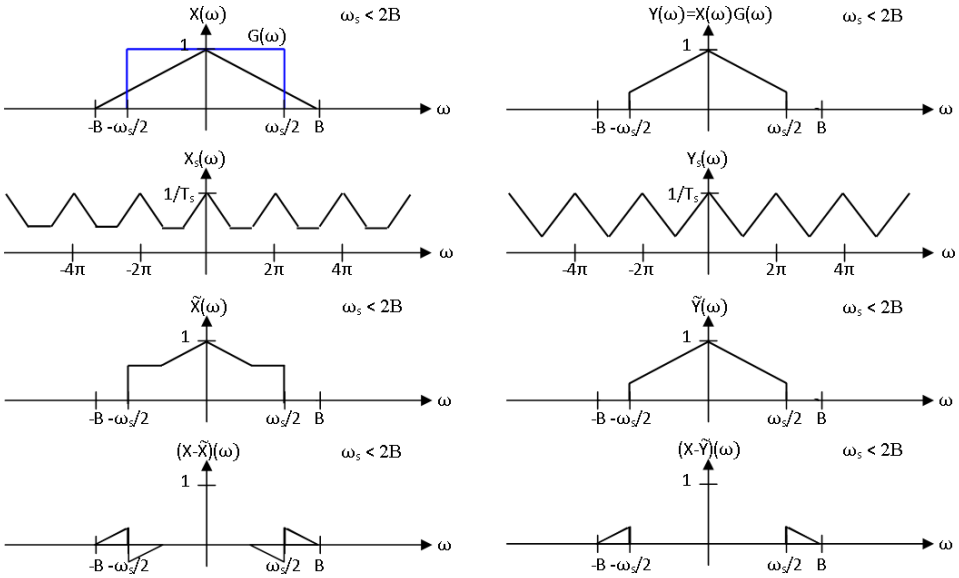


Figure 10.6.1: The figure above illustrates the use of an anti-aliasing filter to improve the process of sampling and reconstruction when using a sampling frequency below the Nyquist frequency. Notice that when using an ideal lowpass anti-aliasing filter, the reconstructed signal spectrum has the same shape as the original signal spectrum for all frequencies below half the sampling rate. This results in a lower error energy when using the anti-aliasing filter, as can be seen by comparing the error spectra shown.

Anti-Aliasing Filters Summary

As can be seen, anti-aliasing filters ensure that the signal is $(-\omega_s/2, \omega_s/2)$ bandlimited, or at least nearly so. The optimal anti-aliasing filter would be the ideal lowpass filter with cutoff frequency at $\omega_s/2$, which would ensure that the original signal spectrum and the reconstructed signal spectrum are equal on the interval $(-\omega_s/2, \omega_s/2)$. However, the ideal lowpass filter is not possible to implement in practice, and approximations must be accepted instead. Anti-aliasing filters are an important component of systems that implement discrete time processing of continuous time signals, as will be shown in the subsequent module.

This page titled 10.6: Anti-Aliasing Filters is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

10.7: Discrete Time Processing of Continuous Time Signals

Introduction

Digital computers can process discrete time signals using extremely flexible and powerful algorithms. However, most signals of interest are continuous time signals, which is how data almost always appears in nature. Now that the theory supporting methods for generating a discrete time signal from a continuous time signal through sampling and then perfectly reconstructing the original signal from its samples without error has been discussed, it will be shown how this can be applied to implement continuous time, linear time invariant systems using discrete time, linear time invariant systems. This is of key importance to many modern technologies as it allows the power of digital computing to be leveraged for processing of analog signals.

Discrete Time Processing of Continuous Time Signals

Process Structure

With the aim of processing continuous time signals using a discrete time system, we will now examine one of the most common structures of digital signal processing technologies. As an overview of the approach taken, the original continuous time signal x is sampled to a discrete time signal x_s in such a way that the periods of the samples spectrum X_s is as close as possible in shape to the spectrum of X. Then a discrete time, linear time invariant filter H_2 is applied, which modifies the shape of the samples spectrum X_s but cannot increase the bandlimit of X_s, to produce another signal y_s. This is reconstructed with a suitable reconstruction filter to produce a continuous time output signal y, thus effectively implementing some continuous time system H_1. This process is illustrated in Figure 10.7.1, and the spectra are shown for a specific case in Figure 10.7.2.

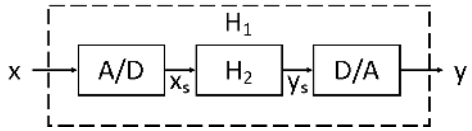


Figure 10.7.1: A block diagram for processing of continuous time signals using discrete time systems is shown.

Further discussion about each of these steps is necessary, and we will begin by discussing the analog to digital converter, often denoted by ADC or A/D. It is clear that in order to process a continuous time signal using discrete time techniques, we must sample the signal as an initial step. This is essentially the purpose of the ADC, although there are practical issues that which will be discussed later. An ADC takes a continuous time analog signal as input and produces a discrete time digital signal as output, with the ideal infinite precision case corresponding to sampling. As stated by the Nyquist-Shannon Sampling theorem, in order to retain all information about the original signal, we usually wish sample above the Nyquist frequency $\omega_s \geq 2B$ where the original signal is bandlimited to $(-B, B)$. When it is not possible to guarantee this condition, an anti-aliasing filter should be used.

The discrete time filter is where the intentional modifications to the signal information occur. This is commonly done in digital computer software after the signal has been sampled by a hardware ADC and before it is used by a hardware DAC to construct the output. This allows the above setup to be quite flexible in the filter that it implements. If sampling above the Nyquist frequency the. Any modifications that the discrete filter makes to this shape can be passed on to a continuous time signal assuming perfect reconstruction. Consequently, the process described will implement a continuous time, linear time invariant filter. This will be explained in more mathematical detail in the subsequent section. As usual, there are, of course, practical limitations that will be discussed later.

Finally, we will discuss the digital to analog converter, often denoted by DAC or D/A. Since continuous time filters have continuous time inputs and continuous time outputs, we must construct a continuous time signal from our filtered discrete time signal. Assuming that we have sampled a bandlimited at a sufficiently high rate, in the ideal case this would be done using perfect reconstruction through the Whittaker-Shannon interpolation formula. However, there are, once again, practical issues that prevent this from happening that will be discussed later.

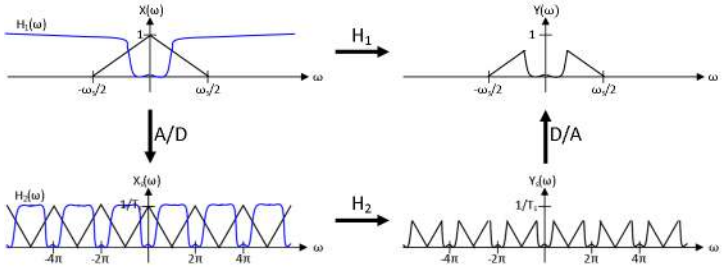


Figure 10.7.2: Spectra are shown in black for each step in implementing a continuous time filter using a discrete time filter for a specific signal. The filter frequency responses are shown in blue, and both are meant to have maximum value 1 in spite of the vertical scale that is meant only for the signal spectra. Ideal ADCs and DACs are assumed.

Discrete Time Filter

With some initial discussion of the process illustrated in Figure 10.7.1 complete, the relationship between the continuous time, linear time invariant filter H_1 and the discrete time, linear time invariant filter H_2 can be explored. We will assume the use of ideal, infinite precision ADCs and DACs that perform sampling and perfect reconstruction, respectively, using a sampling rate $\omega_s = 2\pi/T_s \geq 2B$ where the input signal x is bandlimited to $(-B, B)$. Note that these arguments fail if this condition is not met and aliasing occurs. In that case, pre-application of an anti-aliasing filter is necessary for these arguments to hold.

Recall that we have already calculated the spectrum X_s of the samples x_s given an input x with spectrum X as

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X\left(\frac{\omega - 2\pi k}{T_s}\right).$$

Likewise, the spectrum Y_s of the samples y_s given an output y with spectrum Y is

$$Y_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} Y\left(\frac{\omega - 2\pi k}{T_s}\right).$$

From the knowledge that $y_s = (H_1 x)_s = H_2(x_s)$, it follows that

$$\sum_{k=-\infty}^{\infty} H_1\left(\frac{\omega - 2\pi k}{T_s}\right) X\left(\frac{\omega - 2\pi k}{T_s}\right) = H_2(\omega) \sum_{k=-\infty}^{\infty} X\left(\frac{\omega - 2\pi k}{T_s}\right).$$

Because X is bandlimited to $(-\pi/T_s, \pi/T_s)$, we may conclude that

$$H_2(\omega) = \sum_{k=-\infty}^{\infty} H_1\left(\frac{\omega - 2\pi k}{T_s}\right) (u(\omega - (2k-1)\pi) - u(\omega - (2k+1)\pi)).$$

More simply stated, H_2 is 2π periodic and $H_2(\omega) = H_1(\omega/T_s)$ for $\omega \in [-\pi, \pi)$.

Given a specific continuous time, linear time invariant filter H_1, the above equation solves the system design problem provided we know how to implement H_2. The filter H_2 must be chosen such that it has a frequency response where each period has the same shape as the frequency response of H_1 on $(-\pi/T_s, \pi/T_s)$. This is illustrated in the frequency responses shown in Figure 10.7.2.

We might also want to consider the system analysis problem in which a specific discrete time, linear time invariant filter H_2 is given, and we wish to describe the filter H_1. There are many such filters, but we can describe their frequency responses on $(-\pi/T_s, \pi/T_s)$ using the above equation. Isolating one period of $H_2(\omega)$ yields the conclusion that $H_1(\omega) = H_2(\omega T_s)$ for $\omega \in (-\pi/T_s, \pi/T_s)$. Because x was assumed to be bandlimited to $(-\pi/T_s, \pi/T_s)$, the value of the frequency response elsewhere is irrelevant.

Practical Considerations

As mentioned before, there are several practical considerations that need to be addressed at each stage of the process shown in Figure 10.7.1. Some of these will be briefly addressed here, and a more complete model of how discrete time processing of continuous time signals appears in Figure 10.7.3.

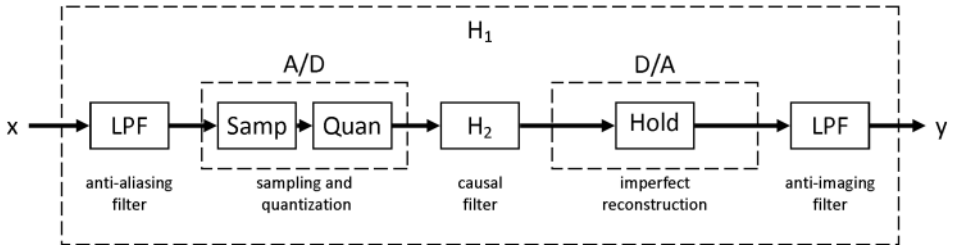


Figure 10.7.3: A more complete model of how discrete time processing of continuous time signals is implemented in practice. Notice the addition of anti-aliasing and anti-imaging filters to promote input and output bandlimitedness. The ADC is shown to perform sampling with quantization. The digital filter is further specified to be causal. The DAC is shown to perform imperfect reconstruction, a zero order hold in this case.

Anti-Aliasing Filter

In reality, we cannot typically guarantee that the input signal will have a specific bandlimit, and sufficiently high sampling rates cannot necessarily be produced. Since it is imperative that the higher frequency components not be allowed to masquerade as lower frequency components through aliasing, anti-aliasing filters with cutoff frequency less than or equal to $\omega_s/2$ must be used before the signal is fed into the ADC. The block diagram in Figure 10.7.3 reflects this addition.

As described in the previous section, an ideal lowpass filter removing all energy at frequencies above $\omega_s/2$ would be optimal. Of course, this is not achievable, so approximations of the ideal lowpass filter with low gain above $\omega_s/2$ must be accepted. This means that some aliasing is inevitable, but it can be reduced to a mostly insignificant level.

Signal Quantization

In our preceding discussion of discrete time processing of continuous time signals, we had assumed an ideal case in which the ADC performs sampling exactly. However, while an ADC does convert a continuous time signal to a discrete time signal, it also must convert analog values to digital values for use in a digital logic device, a phenomenon called quantization. The ADC subsystem of the block diagram in Figure 10.7.3 reflects this addition.

The data obtained by the ADC must be stored in finitely many bits inside a digital logic device. Thus, there are only finitely many values that a digital sample can take, specifically $2N$ where N is the number of bits, while there are uncountably many values an analog sample can take. Hence something must be lost in the quantization process. The result is that quantization limits both the range and precision of the output of the ADC. Both are finite, and improving one at constant number of bits requires sacrificing quality in the other.

Filter Implementability

In real world circumstances, if the input signal is a function of time, the future values of the signal cannot be used to calculate the output. Thus, the digital filter H_2 and the overall system H_1 must be causal. The filter annotation in Figure 10.7.3 reflects this addition. If the desired system is not causal but has impulse response equal to zero before some time t_0, a delay can be introduced to make it causal. However, if this delay is excessive or the impulse response has infinite length, a windowing scheme becomes necessary in order to practically solve the problem. Multiplying by a window to decrease the length of the impulse response can reduce the necessary delay and decrease computational requirements.

Take, for instance the case of the ideal lowpass filter. It is acausal and infinite in length in both directions. Thus, we must satisfy ourselves with an approximation. One might suggest that these approximations could be achieved by truncating the sinc impulse response of the lowpass filter at one of its zeros, effectively windowing it with a rectangular pulse. However, doing so would produce poor results in the frequency domain as the resulting convolution would significantly spread the signal energy. Other windowing functions, of which there are many, spread the signal less in the frequency domain and are thus much more useful for producing these approximations.

Anti-Imaging Filter

In our preceding discussion of discrete time processing of continuous time signals, we had assumed an ideal case in which the DAC performs perfect reconstruction. However, when considering practical matters, it is important to remember that the sinc function,

which is used for Whittaker-Shannon interpolation, is infinite in length and acausal. Hence, it would be impossible for an DAC to implement perfect reconstruction.

Instead, the DAC implements a causal zero order hold or other simple reconstruction scheme with respect to the sampling rate ω_s used by the ADC. However, doing so will result in a function that is not bandlimited to $(-\omega_s/2, \omega_s/2)$. Therefore, an additional lowpass filter, called an anti-imaging filter, must be applied to the output. The process illustrated in Figure 10.7.3 reflects these additions. The anti-imaging filter attempts to bandlimit the signal to $(-\omega_s/2, \omega_s/2)$, so an ideal lowpass filter would be optimal. However, as has already been stated, this is not possible. Therefore, approximations of the ideal lowpass filter with low gain above $\omega_s/2$ must be accepted. The anti-imaging filter typically has the same characteristics as the anti-aliasing filter.

Discrete Time Processing of Continuous Time Signals Summary

As has been show, the sampling and reconstruction can be used to implement continuous time systems using discrete time systems, which is very powerful due to the versatility, flexibility, and speed of digital computers. However, there are a large number of practical considerations that must be taken into account when attempting to accomplish this, including quantization noise and anti-aliasing in the analog to digital converter, filter implementability in the discrete time filter, and reconstruction windowing and associated issues in the digital to analog converter. Many modern technologies address these issues and make use of this process.

This page titled 10.7: Discrete Time Processing of Continuous Time Signals is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

11: Laplace Transform and Continuous Time System Design

11.1: Laplace Transform
11.2: Common Laplace Transforms
11.3: Properties of the Laplace Transform
11.4: Inverse Laplace Transform
11.5: Poles and Zeros in the S-Plane
11.6: Region of Convergence for the Laplace Transform
11.7: Rational Functions and the Laplace Transform
11.8: Differential Equations
11.9: Continuous Time Filter Design

This page titled 11: Laplace Transform and Continuous Time System Design is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

11.1: Laplace Transform

Introduction

The Laplace transform is a generalization of the Continuous-Time Fourier Transform (Section 8.2). It is used because the CTFT does not converge/exist for many important signals, and yet it does for the Laplace-transform (e.g., signals with infinite l_2 norm). It is also used because it is notationaly cleaner than the CTFT. However, instead of using complex exponentials (Section 7.2) of the form $e^{j\omega t}$, with purely imaginary parameters, the Laplace transform uses the more general, e^{st}, where $s = \sigma + j\omega$ is complex, to analyze signals in terms of exponentially weighted sinusoids.

The Laplace Transform

Bilateral Laplace Transform Pair

Although Laplace transforms are rarely solved in practice using integration (tables (Section 11.2) and computers (e.g. Matlab) are much more common), we will provide the **bilateral Laplace transform pair** here for purposes of discussion and derivation. 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 5.1).

Laplace Transform

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

Inverse Laplace Transform

$$f(t) = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty} F(s)e^{st}ds$$

Note

We have defined the bilateral Laplace transform. There is also a **unilateral Laplace transform** ,

$$F(s) = \int_{0}^{\infty} f(t)e^{-(st)}\mathrm{d}t$$

which is useful for solving the difference equations with nonzero initial conditions. This is similar to the **unilateral Z Transform** in Discrete time.

Relation between Laplace and CTFT

Taking a look at the equations describing the Z-Transform and the Discrete-Time Fourier Transform:

Continuous-Time Fourier Transform

$$\mathcal{F}(\Omega) = \int_{-\infty}^{\infty} f(t)e^{-(j\Omega t)}dt$$

Laplace Transform

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

We can see many similarities; first, that :

$$\mathcal{F}(\Omega) = F(s)$$

for all $\Omega = s$.

Note

The CTFT is a complex-valued function of a real-valued variable ω (and 2π periodic). The Z-transform is a complex-valued function of a complex valued variable z.

Plots

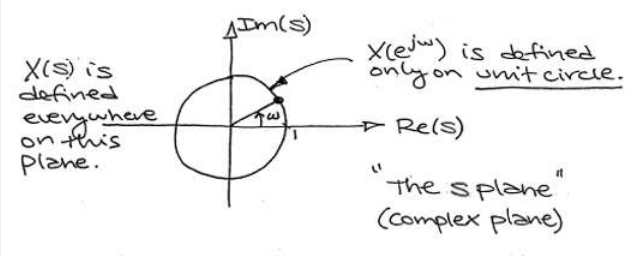
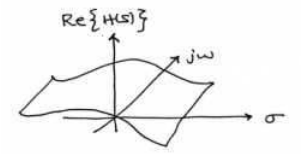


Figure 11.1.1

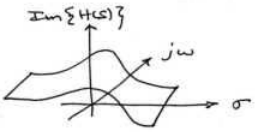
Visualizing the Laplace Transform

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.

Real and imaginary sample plots



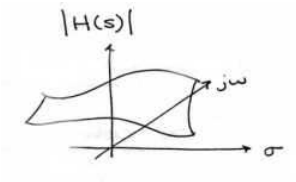
(a) The Real part of $H(s)$



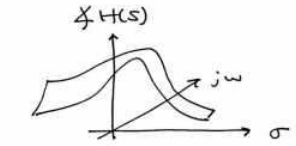
(b) The Imaginary part of $H(s)$

Figure 11.1.2: Real and imaginary parts of $H(s)$ are now each 3-dimensional surfaces.

Magnitude and phase sample plots



(a) The Magnitude of $H(s)$



(b) The Phase of $H(s)$

Figure 11.1.3: Magnitude and phase of $H(s)$ are also each 3-dimensional surfaces. This representation is more common than real and imaginary parts.

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 is much easier to understand and is the way both the Laplace transform and its discrete-time counterpart the Z-transform are represented graphically.

Using a Computer to find the Laplace Transform

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.

Laplace Transform Definition Demonstration

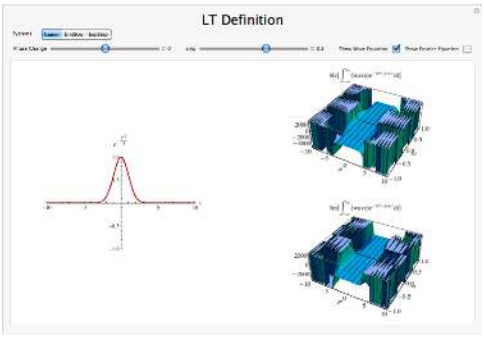


Figure 11.1.4: Interact (when online) with a Mathematica CDF demonstrating the Laplace Transform. To Download, right-click and save target as .cdf.

Khan Lecture on Laplace

Figure 11.1.5: See the attached video on the basics of the Unilateral Laplace Transform from Khan Academy

Conclusion

The laplace transform proves a useful, more general form of the Continuous Time Fourier Transform. It applies equally well to describing systems as well as signals using the eigenfunction method, and to describing a larger class of signals better described using the pole-zero method.

This page titled 11.1: Laplace Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

11.2: Common Laplace Transforms

Table 11.2.1

Signal	Laplace Transform	Region of Convergence
$\delta(t)$	1	All s
$\delta(t-T)$	$e^{(-sT)}$	All s
$u(t)$	$\frac{1}{s}$	$\mathrm{Re}(s) > 0$
$u(-t)$	$\frac{1}{s}$	$\mathrm{Re}(s) < 0$
$tu(t)$	$\frac{1}{s^2}$	$\mathrm{Re}(s) > 0$
$t^n u(t)$	$\frac{n!}{s^{n+1}}$	$\mathrm{Re}(s) > 0$
$-(t^n u(-t))$	$\frac{n!}{s^{n+1}}$	$\mathrm{Re}(s) < 0$
$e^{-(\lambda t)}u(t)$	$\frac{1}{s+\lambda}$	$\mathrm{Re}(s) > -\lambda$
$\left(-e^{-(\lambda t)}\right)u(-t)$	$\frac{1}{s+\lambda}$	$\mathrm{Re}(s) < -\lambda$
$te^{-(\lambda t)}u(t)$	$\frac{1}{(s+\lambda)^2}$	$\mathrm{Re}(s) > -\lambda$
$t^n e^{-(\lambda t)}u(t)$	$\frac{n!}{(s+\lambda)^{n+1}}$	$\mathrm{Re}(s) > -\lambda$
$-\left(t^n e^{-(\lambda t)}u(-t)\right)$	$\frac{n!}{(s+\lambda)^{n+1}}$	$\mathrm{Re}(s) < -\lambda$
$\cos(bt)u(t)$	$\frac{s}{s^2+b^2}$	$\mathrm{Re}(s) > 0$
$\sin(bt)u(t)$	$\frac{b}{s^2+b^2}$	$\mathrm{Re}(s) > 0$
$e^{-(at)}\cos(bt)u(t)$	$\frac{s+a}{(s+a)^2+b^2}$	$\mathrm{Re}(s) > -a$
$e^{-(at)}\sin(bt)u(t)$	$\frac{b}{(s+a)^2+b^2}$	$\mathrm{Re}(s) > -a$
$\frac{d^n}{dt^n}\delta(t)$	s^n	All s

This page titled 11.2: Common Laplace Transforms is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

11.3: Properties of the Laplace Transform

Table 11.3.1: Table of Laplace Transform Properties.

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 $\mathrm{ROC}_1 \cap \mathrm{ROC}_2$
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	$x(\alpha t)$	$(1-\lvert\alpha\rvert)X(s-\alpha)$	Scaled ROC ($s-\alpha$ must be in the region of convergence)
Conjugation	$x^*(t)$	$X^*(s^*)$	ROC
Convolution	$x_1(t) * x_2(t)$	$X_1(t)X_2(t)$	At least $\mathrm{ROC}_1 \cap \mathrm{ROC}_2$
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 $\mathrm{ROC} \cap (\mathrm{Re}(s) > 0)$

This page titled 11.3: Properties of the Laplace Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

11.4: Inverse Laplace Transform

Introduction

When using the Laplace-transform (Section 11.1)

$$H(s) = \sum_{t=-\infty}^{\infty} h(t)s^{-t}$$

it is often useful to be able to find $h(t)$ given $H(s)$. There are at least 4 different methods to do this:

1. Inspection
2. Partial-Fraction Expansion
3. Power Series Expansion
4. Contour Integration

Inspection Method

This "method" is to basically become familiar with the Laplace-transform pair tables (Section 11.2) and then "reverse engineer".

Example 11.4.1

When given

$$H(s) = \frac{s}{s-\alpha}$$

with an ROC (Section 12.6) of

$$|s| > \alpha$$

we could determine "by inspection" that

$$h(t) = \alpha^t u(t)$$

Partial-Fraction Expansion Method

When dealing with **linear time-invariant systems** the z-transform is often of the form

$$\begin{aligned} H(s) &= \frac{B(s)}{A(s)} \\ &= \frac{\sum_{k=0}^{M} b_k s^{-k}}{\sum_{k=0}^{N} a_k s^{-k}} \end{aligned} \tag{11.4.1}$$

This can also expressed as

$$H(s) = \frac{a_0}{b_0} \frac{\prod_{k=1}^{M} 1 - c_k s^{-1}}{\prod_{k=1}^{N} 1 - d_k s^{-1}}$$

where c_k represents the nonzero zeros of $H(s)$ and d_k represents the nonzero poles.

If $M < N$ then $H(s)$ can be represented as

$$H(s) = \sum_{k=1}^{N} \frac{A_k}{1 - d_k s^{-1}}$$

This form allows for easy inversions of each term of the sum using the inspection method and the transform table. If the numerator is a polynomial, however, then it becomes necessary to use partial-fraction expansion to put $H(s)$ in the above form. If $M \geq N$ then $H(s)$ can be expressed as

$$H(s) = \sum_{r=0}^{M-N} B_r s^{-r} + \frac{\sum_{k=0}^{N-1} b'_k s^{-k}}{\sum_{l=0}^{N} a_k s^{-k}}$$

Example 11.4.2

Find the inverse z-transform of

$$H(s) = \frac{1 + 2s^{-1} + s^{-2}}{1 - 3s^{-1} + 2s^{-2}}$$

where the ROC is $|s| > 2$. In this case $M = N = 2$, so we have to use long division to get

$$H(s) = \frac{1}{2} + \frac{\frac{1}{2} + \frac{7}{2}s^{-1}}{1 - 3s^{-1} + 2s^{-2}}$$

Next factor the denominator.

$$H(s) = 2 + \frac{-1 + 5s^{-1}}{(1 - 2s^{-1})(1 - s^{-1})}$$

Now do partial-fraction expansion.

$$H(s) = \frac{1}{2} + \frac{A_1}{1 - 2s^{-1}} + \frac{A_2}{1 - s^{-1}} = \frac{1}{2} + \frac{\frac{9}{2}}{1 - 2s^{-1}} + \frac{-4}{1 - s^{-1}}$$

Now each term can be inverted using the inspection method and the Laplace-transform table. Thus, since the ROC is $|s| > 2$,

$$h(t) = \frac{1}{2}\delta(t) + \frac{9}{2}2^t u(t) - 4u(t)$$

Demonstration of Partial Fraction Expansion

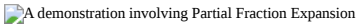


Figure 11.4.1: Interactive experiment illustrating how the Partial Fraction Expansion method is used to solve a variety of numerator and denominator problems. (To view and interact with the simulation, download the free Mathematica player at www.wolfram.com/products/player/download.cgi)

Khan Lecture on Partial Fraction Expansion

Figure 11.4.2: video from Khan Academy

Power Series Expansion Method

When the z-transform is defined as a power series in the form

$$H(s) = \sum_{t=-\infty}^{\infty} h(t)s^{-t}$$

then each term of the sequence $h(t)$ can be determined by looking at the coefficients of the respective power of s^{-t}.

Example 11.4.3

Now look at the Laplace-transform of a **finite-length sequence**.

$$\begin{aligned} H(s) &= s^2\left(1 + 2s^{-1}\right)\left(1 - \frac{1}{2}s^{-1}\right)\left(1 + s^{-1}\right) \\ &= s^2 + \frac{5}{2}s + \frac{1}{2} + -s^{-1} \end{aligned} \tag{11.4.2}$$

In this case, since there were no poles, we multiplied the factors of $H(s)$. Now, by inspection, it is clear that

$$h(t) = \delta(t+2) + \frac{5}{2}\delta(t+1) + \frac{1}{2}\delta(t) + -\delta[t-1].$$

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 11.4.4

Suppose

$$H(s) = \log_t \left(1 + \alpha s^{-1}\right)$$

Noting that

$$\log_t(1+x) = \sum_{t=1}^{\infty} \frac{-1^{t+1} x^t}{t}$$

Then

$$H(s) = \sum_{t=1}^{\infty} \frac{-1^{t+1} \alpha^t s^{-t}}{t}$$

Therefore

$$H(s) = \begin{cases} \frac{-1^{t+1}\alpha^t}{t} \text{ if } t \geq 1 \\ 0 \text{ if } t \leq 0 \end{cases}$$

Contour Integration Method

Without going in to much detail

$$h(t) = \frac{1}{2\pi i} \oint_r H(s) s^{t-1} \mathrm{d}s$$

where r is a counter-clockwise contour in the ROC of $H(s)$ encircling the origin of the s-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.

Demonstration of Contour Integration

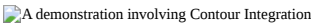


Figure 11.4.3: Interactive experiment illustrating how the contour integral is applied on a simple example. For a more in-depth discussion of this method, some background in complex analysis is required. (To view and interact with the simulation, download the free Mathematica player at www.wolfram.com/products/player/download.cgi)

Conclusion

The Inverse Laplace-transform is very useful to know for the purposes of designing a filter, and there are many ways in which to calculate it, drawing from many disparate areas of mathematics. All nevertheless assist the user in reaching the desired time-domain signal that can then be synthesized in hardware(or software) for implementation in a real-world filter.

This page titled 11.4: Inverse Laplace Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

11.5: Poles and Zeros in the S-Plane

Introduction to Poles and Zeros of the Laplace-Transform

It is quite difficult to qualitatively analyze the Laplace transform (Section 11.1) and Z-transform, 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's poles and zeros to try to gain a qualitative idea of what a system does.

Once the Laplace-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 Laplace-transform will have the below structure, based on Rational Functions (Section 12.7):

$$H(s) = \frac{P(s)}{Q(s)}$$

The two polynomials, $P(s)$ and $Q(s)$, allow us to find the **poles and zeros** of the Laplace-Transform.

Definition: zeros

1. The value(s) for ss where $P(s) = 0$.
2. The complex frequencies that make the overall gain of the filter transfer function zero.

Definition: poles

1. The value(s) for s where $Q(s) = 0$.
2. The complex frequencies that make the overall gain of the filter transfer function infinite.

Example 11.5.1

Below is a simple transfer function with the poles and zeros shown below it.

$$H(s) = \frac{s+1}{\left(s-\frac{1}{2}\right)\left(s+\frac{3}{4}\right)}$$

The zeros are: $\{-1\}$

The poles are: $\left\{\frac{1}{2}, -\frac{3}{4}\right\}$

The S-Plane

Once the poles and zeros have been found for a given Laplace Transform, they can be plotted onto the S-Plane. The S-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 S-Plane, and examples of plotting zeros and poles onto the plane can be found in the following section.

S-Plane

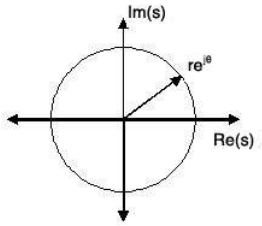


Figure 11.5.1

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 S-Plane.

Example 11.5.2: Simple Pole/Zero Plot

$$H(s) = \frac{s}{\left(s - \frac{1}{2}\right)\left(s + \frac{3}{4}\right)}$$

The zeros are: $\{0\}$

The poles are: $\left\{\frac{1}{2}, -\frac{3}{4}\right\}$

Pole/Zero Plot

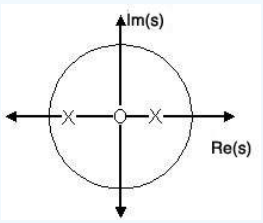


Figure 11.5.2: Using the zeros and poles found from the transfer function, the one zero is mapped to zero and the two poles are placed at $\frac{1}{2}$ and $-\frac{3}{4}$

Example 11.5.3: Complex Pole/Zero Plot

$$H(s) = \frac{(s-j)(s+j)}{\left(s - \left(\frac{1}{2} - \frac{1}{2}j\right)\right)\left(s - \frac{1}{2} + \frac{1}{2}j\right)}$$

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

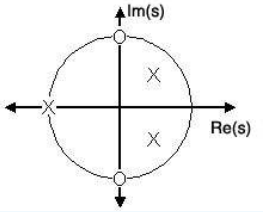


Figure 11.5.3: Using the zeros and poles found from the transfer function, the zeros are mapped to $\pm(j)$, and the poles are placed at -1, $\frac{1}{2} + \frac{1}{2}j$, and $\frac{1}{2} - \frac{1}{2}j$

Example 11.5.4: 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.

Note: 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.

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');
```

Interactive Demonstration of Poles and Zeros

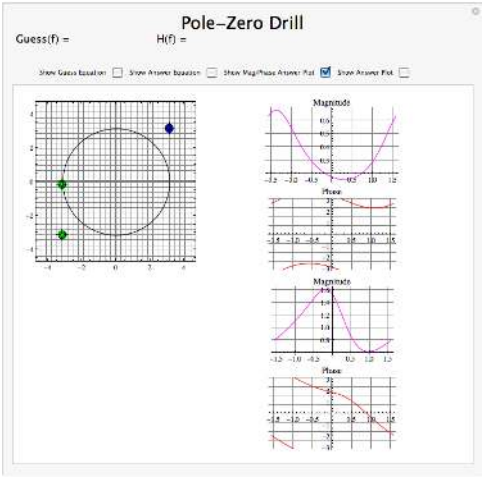


Figure 11.5.4: Interact (when online) with a Mathematica CDF demonstrating Pole/Zero Plots. To Download, right-click and save target as .cdf.

Applications for pole-zero plots

Stability and Control theory

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 3.6).

Pole/Zero Plots and the 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 11.5.2 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 11.5.5

$$H(z) = \frac{z}{\left(z - \frac{1}{2}\right)\left(z + \frac{3}{4}\right)}$$

Region of Convergence for the Pole/Zero Plot

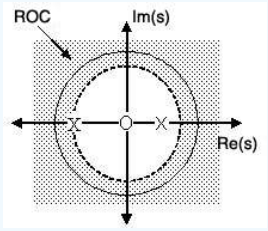


Figure 11.5.5: The shaded area represents the chosen ROC for the transfer function.

Frequency Response and Pole/Zero Plots

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.

Conclusion

Pole-Zero Plots are clearly quite useful in the study of the Laplace and Z transform, affording us a method of visualizing the at times confusing mathematical functions.

This page titled 11.5: Poles and Zeros in the S-Plane is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

11.6: Region of Convergence for the Laplace Transform

With the Laplace transform (Section 11.1), the s-plane represents a set of signals (complex exponentials (Section 1.8)). 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

$$H(s) = \int_{-\infty}^{\infty} h(t)e^{-(st)}\mathrm{d}t$$

If we consider a causal (Section 1.1), complex exponential, $h(t) = e^{-(at)}u(t)$, we get the equation,

$$\int_{0}^{\infty} e^{-(at)}e^{-(st)}dt = \int_{0}^{\infty} e^{-((a+s)t)}dt$$

Evaluating this, we get

$$\frac{-1}{s+a}\left(\lim_{t\to\infty} e^{-((s+a)t)} - 1\right)$$

Notice that this equation will tend to infinity when $\lim_{t\to\infty} e^{-((s+a)t)}$ tends to infinity. To understand when this happens, we take one more step by using $s = \sigma + j\omega$ to realize this equation as

$$\lim_{t\to\infty} e^{-(j\omega t)}e^{-((\sigma+a)t)}$$

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 tt tends to infinity. On the other hand, if σ+aσ 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

$$\mathrm{Re}(s) > -a$$

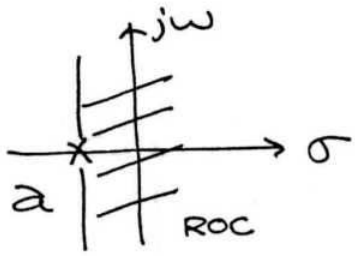
Although we will not go through the process again for anticausal signals, we could. In doing so, we would find that the necessary condition for convergence is when

Necessary Condition for Anti-Causal Convergence

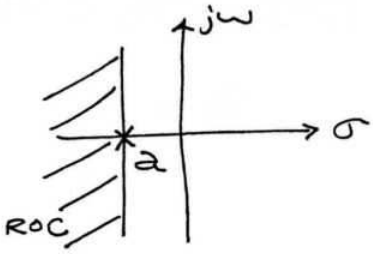
$$\mathrm{Re}(s) < -a$$

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.



(a) The Region of Convergence for a causal signal.



b) The Region of Convergence for an anti-causal signal.

Figure 11.6.1

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 11.6.1

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

$$H_1(s) = \frac{1}{s+a},\ \mathrm{Re}(s) > -a$$
$$H_2(s) = \frac{-1}{s+b},\ \mathrm{Re}(s) < -b$$

$-b > \mathrm{Re}(s) > -a$. If $a > b$, we can represent this graphically. Otherwise, there will be no region of convergence.

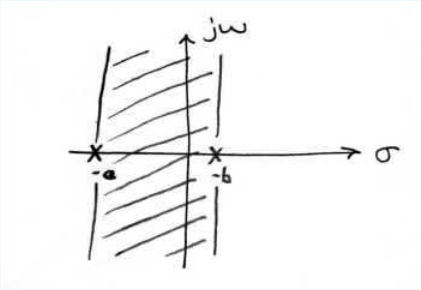


Figure 11.6.2: The Region of Convergence of $h(t)$ if $a > b$.

This page titled 11.6: Region of Convergence for the Laplace Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

11.7: Rational Functions and the Laplace Transform

Introduction

When dealing with operations on polynomials, the term **rational function** is a simple way to describe a particular relationship between two polynomials.

Definition: Rational Function

For any two polynomials, A and B, their quotient is called a rational function.

Example 11.7.1

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.

$$f(x) = \frac{x^2 - 4}{2x^2 + x - 3} \tag{11.7.1}$$

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 to see how rational functions are useful when dealing with the Laplace transform.

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, Equation 11.7.1.

$$f(x) = \frac{(x+2)(x-2)}{(2x+3)(x-1)}$$

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.

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.

Example 11.7.2

Continuing to look at our rational function above, Equation 11.7.1, 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.

Domain

Using the roots that we found above, the **domain** of the rational function can be easily defined.

Definition: Domain

The group, or set, of values that are defined by a given function.

Example 11.7.3

Using the rational function above, Equation 11.7.1, 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:

$$\{x \in \mathbb{R} | (x \neq -3) \text{ and } (x \neq 1)\}$$

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.

Rational Functions and the Laplace Transform

Rational functions often result when the Laplace transform is used to compute transfer functions for LTI systems. When using the Laplace transform to solve linear constant coefficient ordinary differential equations, partial fraction expansions of rational functions prove particularly useful. The roots of the polynomials in the numerator and denominator of the transfer function play an important role in describing system behavior. The roots of the polynomial in the numerator produce zeros of the transfer function where the system produces no output for an input of that complex frequency. The roots of the polynomial in the denominator produce poles of the transfer function where the system has natural frequencies of oscillation.

Summary

Once we have used our knowledge of rational functions to find its roots, we can manipulate a Laplace transform in a number of useful ways. We can apply this knowledge by representing an LTI system graphically through a pole-zero plot for analysis or design.

This page titled 11.7: Rational Functions and the Laplace Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

11.8: Differential Equations

Differential Equations

It is often useful to describe systems using equations involving the rate of change in some quantity through differential equations. Recall that one important subclass of differential equations, linear constant coefficient ordinary differential equations, takes the form

$$Ay(t) = x(t) \tag{11.8.1}$$

where A is a differential operator of the form

$$A = a_n \frac{d^n}{dt^n} + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} + ... + a_1 \frac{d}{dt} + a_0 \tag{11.8.2}$$

The differential equation in Equation 11.8.1 would describe some system modeled by A with an input forcing function $x(t)$ that produces an output solution signal $y(t)$. However, the unilateral Laplace transform permits a solution for initial value problems to be found in what is usually a much simpler method. Specifically, it greatly simplifies the procedure for nonhomogeneous differential equations.

General Formulas for the Differential Equation

As stated briefly in the definition above, a differential equation is a very useful tool in describing and calculating the change in an output of a system described by the formula for a given input. The key property of the differential equation is its ability to help easily find the transform, $H(s)$, of a system. In the following two subsections, we will look at the general form of the differential equation and the general conversion to a Laplace-transform directly from the differential equation.

Conversion to Laplace-Transform

We can easily generalize the **transfer function**, $H(s)$, for any differential equation. Below are the steps taken to convert any differential equation into its transfer function, i.e. Laplace-transform. The first step involves taking the Fourier Transform of all the terms in [link]. Then we use the linearity property to pull the transform inside the summation and the time-shifting property of the Laplace-transform to change the time-shifting terms to exponentials. Once this is done, we arrive at the following equation: $a_0 = 1$.

$$Y(s) = -\sum_{k=1}^{N} a_k Y(s) s^{-k} + \sum_{k=0}^{M} b_k X(s) s^{-k}$$

$$\begin{aligned} H(s) &= \frac{Y(s)}{X(s)} \\ &= \frac{\sum_{k=0}^{M} b_k s^{-k}}{1 + \sum_{k=1}^{N} a_k s^{-k}} \end{aligned} \tag{11.8.3}$$

Conversion to Frequency Response

Once the Laplace-transform has been calculated from the differential equation, we can go one step further to define the frequency response of the system, or filter, that is being represented by the differential 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 Laplace-transform. The conversion is simply a matter of taking the Laplace-transform formula, $H(s)$, and replacing every instance of s with e^{iw}.

$$H(w) = H(s)|_{s,s=e^{jw}} = \frac{\sum_{k=0}^{M} b_k e^{-(jwk)}}{\sum_{k=0}^{N} a_k e^{-(jwk)}} \tag{11.8.4}$$

Once you understand the derivation of this formula, look at the module concerning Filter Design from the Laplace-Transform (Section 12.9) for a look into how all of these ideas of the Laplace-transform (Section 11.1), Differential Equation, and Pole/Zero Plots (Section 12.5) play a role in filter design.

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, $x(t)$, and a set of initial conditions. Two common methods exist for solving a LCCDE: the **direct method** and the **indirect method**, the latter being based on the Laplace-transform. Below we will briefly discuss the formulas for solving a LCCDE using each of these methods.

Direct Method

The final solution to the output based on the direct method is the sum of two parts, expressed in the following equation:

$$y(t) = y_h(t) + y_p(t)$$

The first part, $y_h(t)$, is referred to as the **homogeneous solution** and the second part, $y_h(t)$, 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.

Homogeneous Solution

We begin by assuming that the input is zero, $x(t) = 0$. Now we simply need to solve the homogeneous differential equation:

$$\sum_{k=0}^{N} a_k y(t-k) = 0$$

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:

$$\sum_{k=0}^{N} a_k \lambda^{t-k} = 0$$

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:

$$y_h(t) = C_1 (\lambda_1)^t + C_2 (\lambda_2)^t + ... + C_N (\lambda_N)^t$$

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:

$$y_h(t) = C_1 (\lambda_1)^t + C_1 t (\lambda_1)^t + C_1 t^2 (\lambda_1)^t + ... + C_1 t^{K-1} (\lambda_1)^t + C_2 (\lambda_2)^t + ... + C_N (\lambda_N)^t$$

Particular Solution

The particular solution, $y_p(t)$, will be any solution that will solve the general differential equation:

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

In order to solve, our guess for the solution to $y_p(t)$ will take on the form of the input, $x(t)$. After guessing at a solution to the above equation involving the particular solution, one only needs to plug the solution into the differential equation and solve it out.

Indirect Method

The indirect method utilizes the relationship between the differential equation and the Laplace-transform, discussed earlier, to find a solution. The basic idea is to convert the differential equation into a Laplace-transform, as described above, to get the resulting output, $Y(s)$. Then by inverse transforming this and using partial-fraction expansion, we can arrive at the solution.

$$L\left\{\frac{d}{dt}y(t)\right\} = sY(s) - y(0)$$

This can be interatively extended to an arbitrary order derivative as in Equation 11.8.5.

$$L\left\{\frac{d^n}{dt^n}y(t)\right\} = s^nY(s) - \sum_{m=0}^{n-1} s^{n-m-1}y^{(m)}(0) \tag{11.8.5}$$

Now, the Laplace transform of each side of the differential equation can be taken

$$L\left\{\sum_{k=0}^{n} a_k \frac{d^k}{dt^k}y(t)\right\} = L\{x(t)\}$$

which by linearity results in

$$\sum_{k=0}^{n} a_k L\left\{\frac{d^k}{dt^k}y(t)\right\} = L\{x(t)\}$$

and by differentiation properties in

$$\sum_{k=0}^{n} a_k \left(s^k L\{y(t)\} - \sum_{m=0}^{k-1} s^{k-m-1}y^{(m)}(0)\right) = L\{x(t)\}.$$

Rearranging terms to isolate the Laplace transform of the output,

$$L\{y(t)\} = \frac{L\{x(t)\} + \sum_{k=0}^{n}\sum_{m=0}^{k-1} a_k s^{k-m-1}y^{(m)}(0)}{\sum_{k=0}^{n} a_k s^k}.$$

Thus, it is found that

$$Y(s) = \frac{X(s) + \sum_{k=0}^{n}\sum_{m=0}^{k-1} a_k s^{k-m-1}y^{(m)}(0)}{\sum_{k=0}^{n} a_k s^k}. \tag{11.8.6}$$

In order to find the output, it only remains to find the Laplace transform $X(s)$ of the input, substitute the initial conditions, and compute the inverse Laplace transform of the result. Partial fraction expansions are often required for this last step. This may sound daunting while looking at Equation 11.8.6, but it is often easy in practice, especially for low order differential equations. Equation 11.8.6 can also be used to determine the transfer function and frequency response.

As an example, consider the differential equation

$$\frac{d^2}{dt^2}y(t) + 4\frac{d}{dt}y(t) + 3y(t) = \cos(t)$$

with the initial conditions $y'(0) = 1$ and $y(0) = 0$. Using the method described above, the Laplace transform of the solution $y(t)$ is given by

$$Y(s) = \frac{s}{(s^2+1)(s+1)(s+3)} + \frac{1}{(s+1)(s+3)}.$$

Performing a partial-fraction decomposition, this also equals

$$Y(s) = 0.25\frac{1}{s+1} - 0.35\frac{1}{s+3} + 0.1\frac{s}{s^2+1} + 0.2\frac{1}{s^2+1}.$$

Computing the inverse Laplace transform,

$$y(t) = \left(0.25e^{-t} - 0.35e^{-3t} + 0.1\cos(t) + 0.2\sin(t)\right)u(t).$$

One can check that this satisfies that this satisfies both the differential equation and the initial conditions.

Summary

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 helps in understanding and manipulating a system.

This page titled 11.8: Differential Equations is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

11.9: Continuous Time Filter Design

Introduction

Analog (Continuous-Time) filters are useful for a wide variety of applications, and are especially useful in that they are very simple to build using standard, passive R,L,C components. Having a grounding in basic filter design theory can assist one in solving a wide variety of signal processing problems.

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:

$$\begin{aligned} H(w) &= H(z)|_{z,z=e^{jw}} \\ &= \frac{\sum_{k=0}^{M} b_k e^{-(jwk)}}{\sum_{k=0}^{N} a_k e^{-(jwk)}} \end{aligned}$$

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:

$$H(w) = \left|\frac{b_0}{a_0}\right| \frac{\prod_{k=1}^{M} |e^{jw} - c_k|}{\prod_{k=1}^{N} |e^{jw} - d_k|} \tag{11.9.1}$$

From Equation 11.9.1 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 pi. These characteristics allow us to interpret $|H(w)|$ as follows:

$$|H(w)| = \left|\frac{b_0}{a_0}\right| \frac{\prod \text{"distances from zeros"}}{\prod \text{"distances from poles"}}$$

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.

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 12.5) module.

Example 11.9.1

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 ww goes from 00 to 2π2, and also show the physical meaning of the terms in Equation 11.9.1 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 Equation 11.9.1.

Figure 11.9.1: (a) Pole/Zero Plot represents the pole/zero plot with a few representative vectors graphed. (b) Frequency Response: $|H(w)|$ figure shows the frequency response with a peak at +2 and graphed between plus and minus π.

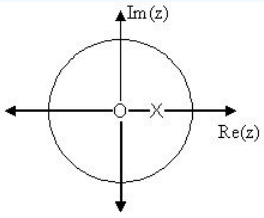
Example 11.9.2

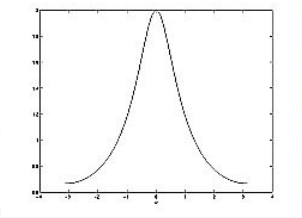
For this example, a more complex transfer function is analyzed in order to represent the system's frequency response.

$$H(z) = \frac{z}{z - \frac{1}{2}} = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

$$H(w) = \frac{1}{1 - \frac{1}{2}e^{-(jw)}}$$

Below we can see the two figures described by the above equations. The Figure $11.9.2(a)$ represents the basic pole/zero plot of the z-transform, $H(w)$. Figure $11.9.2(b)$ 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 Equation 11.9.1 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.





Copy and Paste Image here. Delete this placeholder image

Copy and Paste Image here. Delete this placeholder image

Figure 11.9.2: The (a) Pole/Zero Plot figure represents the pole/zero plot while the Frequency Response ($|H(w)|$) shows the frequency response with a peak at +2 and graphed between plus and minus π.

Types of Filters

Butterworth Filters

The Butterworth filter is the simplest filter. It can be constructed out of passive R, L, C circuits. The magnitude of the transfer function for this filter is

Magnitude of Butterworth Filter Transfer Function

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}}$$

where n is the **order** of the filter and ω_c is the **cutoff frequency**. The cutoff frequency is the frequency where the magnitude experiences a 3 dB dropoff (where $|H(j\omega)| = \frac{1}{\sqrt{2}}$).

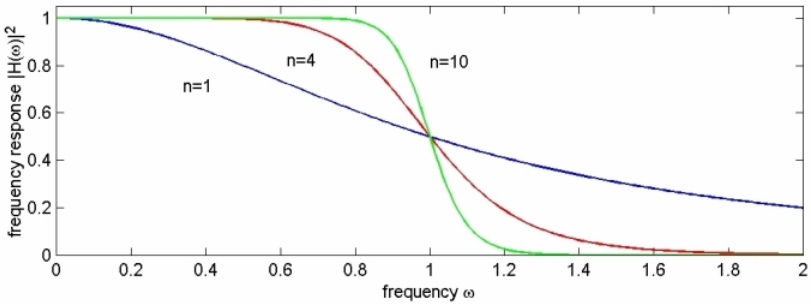


Figure 11.9.3: Three different orders of lowpass Butterworth analog filters: $n = 1, 4, 10$. As n increases, the filter more closely approximates an ideal brickwall lowpass response.

The important aspects of Figure 11.9.3 are that it does not ripple in the passband or stopband as other filters tend to, and that the larger nn, the sharper the cutoff (the smaller the transition band).

Butterworth filters give transfer functions ($H(j\omega)$ and $H(s)$) that are **rational functions**. They also have only poles, resulting in a transfer function of the form

$$\frac{1}{(s-s_1)(s-s_2)\cdots(s-s_n)}$$

and a pole-zero plot of

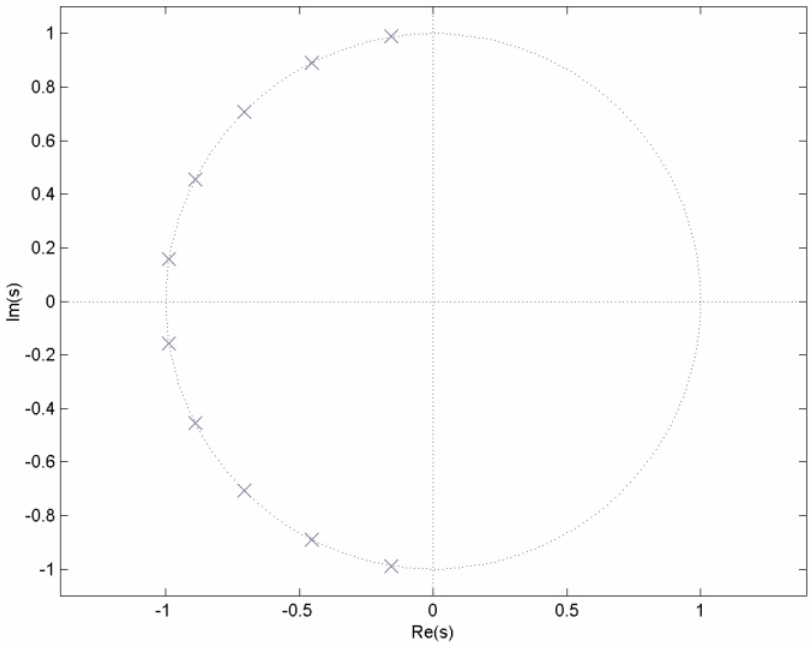


Figure 11.9.4: Poles of a 10th-order ($n = 5$) lowpass Butterworth filter.

Note that the poles lie along a circle in the s-plane.

Chebyshev Filters

The Butterworth filter does not give a sufficiently good approximation across the complete passband in many cases. The Taylor's series approximation is often not suited to the way specifications are given for filters. An alternate error measure is the maximum of the absolute value of the difference between the actual filter response and the ideal. This is considered over the total passband. This is the Chebyshev error measure and was defined and applied to the FIR filter design problem. For the IIR filter, the Chebyshev error is minimized over the passband and a Taylor's series approximation at $\omega = \infty$ is used to determine the stopband performance. This mixture of methods in the IIR case is called the Chebyshev filter, and simple design formulas result, just as for the Butterworth filter.

The design of Chebyshev filters is particularly interesting, because the results of a very elegant theory insure that constructing a frequency-response function with the proper form of equal ripple in the error will result in a minimum Chebyshev error without explicitly minimizing anything. This allows a straightforward set of design formulas to be derived which can be viewed as a generalization of the Butterworth formulas.

The form for the magnitude squared of the frequency-response function for the Chebyshev filter is

$$|F(j\omega)|^2 = \frac{1}{1+\epsilon^2 C_N(\omega)^2} \tag{11.9.2}$$

where $C_N(\omega)$ is an Nth-order Chebyshev polynomial and ϵ is a parameter that controls the ripple size. This polynomial in ω has very special characteristics that result in the optimality of the response function (Equation 11.9.2).

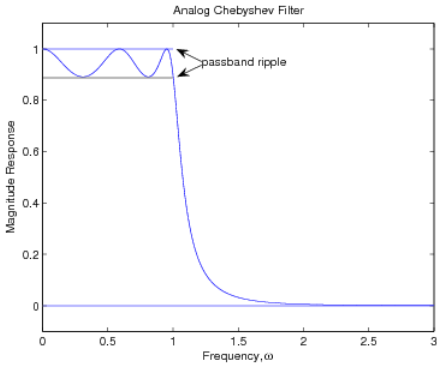


Figure 11.9.5: Fifth Order Chebyshev Filter Frequency Response

Bessel filters

Insert bessel filter information

Elliptic Filters

There is yet another method that has been developed that uses a Chebyshev error criterion in both the passband and the stopband. This is the fourth possible combination of Chebyshev and Taylor's series approximations in the passband and stopband. The resulting filter is called an elliptic-function filter, because elliptic functions are normally used to calculate the pole and zero locations. It is also sometimes called a Cauer filter or a rational Chebyshev filter, and it has equal ripple approximation error in both pass and stopbands.

The error criteria of the elliptic-function filter are particularly well suited to the way specifications for filters are often given. For that reason, use of the elliptic-function filter design usually gives the lowest order filter of the four classical filter design methods for a given set of specifications. Unfortunately, the design of this filter is the most complicated of the four. However, because of the efficiency of this class of filters, it is worthwhile gaining some understanding of the mathematics behind the design procedure.

This section sketches an outline of the theory of elliptic- function filter design. The details and properties of the elliptic functions themselves should simply be accepted, and attention put on understanding the overall picture. A more complete development is available in.

Because both the passband and stopband approximations are over the entire bands, a transition band between the two must be defined. Using a normalized passband edge, the bands are defined by

$$\begin{aligned} 0 < \omega < 1 \quad & \text{passband} \\ 1 < \omega < \omega_s \quad & \text{transition band} \\ \omega_s < \omega < \infty \quad & \text{stopband} \end{aligned}$$

This is illustrated in Figure 11.9.6.

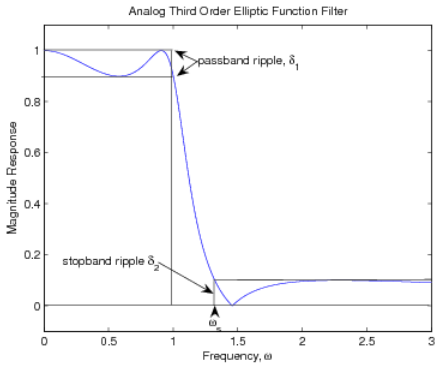


Figure 11.9.6: Third Order Analog Elliptic Function Lowpass Filter showing the Ripples and Band Edges

The characteristics of the elliptic function filter are best described in terms of the four parameters that specify the frequency response:

1. The maximum variation or ripple in the passband δ_1,
2. The width of the transition band $(\omega_s - 1)$,
3. The maximum response or ripple in the stopband δ_2, and
4. The order of the filter N.

The result of the design is that for any three of the parameters given, the fourth is minimum. This is a very flexible and powerful description of a filter frequency response.

The form of the frequency-response function is a generalization of that for the Chebyshev filter

$$FF(j\omega) = |F(j\omega)|^2 = \frac{1}{1 + \epsilon^2 G^2(\omega)}$$

where

$$FF(s) = F(s)F(-s)$$

with $F(s)$ being the prototype analog filter transfer function similar to that for the Chebyshev filter. $G(\omega)$ is a rational function that approximates zero in the passband and infinity in the stopband. The definition of this function is a generalization of the definition of the Chebyshev polynomial.

Filter Design Demonstration

Conclusion

As can be seen, there is a large amount of information available in filter design, more than an introductory module can cover. Even for designing Discrete-time IIR filters, it is important to remember that there is a far larger body of literature for design methods for the analog signal processing world than there is for the digital. Therefore, it is often easier and more practical to implement an IIR filter using standard analog methods, and then discretize it using methods such as the Bilateral Transform.

This page titled 11.9: Continuous Time Filter Design is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

12: Z-Transform and Discrete Time System Design

12.1: Z-Transform
12.2: Common Z-Transforms
12.3: Properties of the Z-Transform
12.4: Inverse Z-Transform
12.5: Poles and Zeros in the Z-Plane
12.6: Region of Convergence for the Z-Transform
12.7: Rational Functions and the Z-Transform
12.8: Difference Equations
12.9: Discrete Time Filter Design

This page titled 12: Z-Transform and Discrete Time System Design is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

12.1: Z-Transform

Introduction

The Z transform is a generalization of the Discrete-Time Fourier Transform (Section 9.2). It is used because the DTFT does not converge/exist for many important signals, and yet does for the z-transform. It is also used because it is notationally cleaner than the DTFT. In contrast to the DTFT, instead of using complex exponentials (Section 7.2) of the form (e^{j \omega n}\), with purely imaginary parameters, the Z transform uses the more general, z^n, where z is complex. The Z-transform thus allows one to bring in the power of complex variable theory into Digital Signal Processing.

The Z-Transform

Bilateral Z-transform Pair

Although Z transforms are rarely solved in practice using integration (tables and computers (e.g. Matlab) are much more common), we will provide the **bilateral Z transform pair** here for purposes of discussion and derivation. These define the forward and inverse Z 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 5.1).

Z Transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

Inverse Z Transform

$$x[n] = \frac{1}{2\pi j} \oint_r X(z) z^{n-1} \mathrm{d}z$$

Note

We have defined the bilateral z-transform. There is also a **unilateral z-transform** ,

$$X(z) = \sum_{n=0}^{\infty} x[n] z^{-n}$$

which is useful for solving the difference equations with nonzero initial conditions. This is similar to the **unilateral Laplace Transform** in continuous time.

Relation between Z-transform and DTFT

Taking a look at the equations describing the Z-Transform and the Discrete-Time Fourier Transform:

Discrete-Time Fourier Transform

$$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x(n) e^{-(j\omega n)}$$

Z-Transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

We can see many similarities; first, that:

$$X\left(e^{j\omega}\right) = X(z)$$

for all $z = e^{j\omega}$

Visualizing the Z-transform

With the DTFT, we have a complex-valued function of a real-valued variable ω (and 2π periodic). The Z-transform is a complex-valued function of a complex valued variable z.

Plots

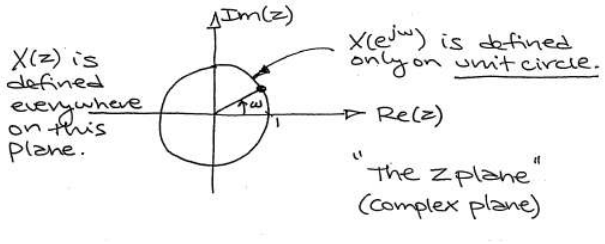


Figure 12.1.1

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 Z, 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.

Consider the z-transform given by $H(z) = z$, as illustrated below.

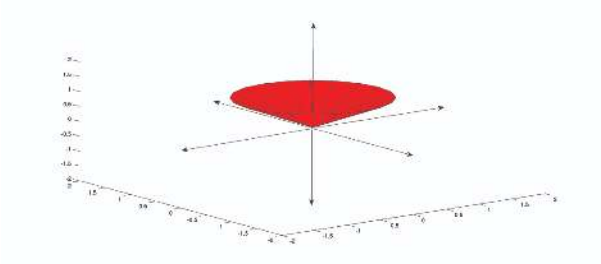
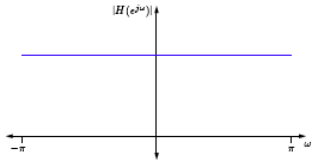


Figure 12.1.2

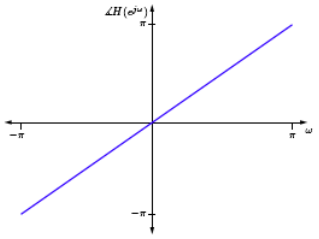
The corresponding DTFT has magnitude and phase given below.

Note

While these are legitimate ways of looking at a signal in the Z 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 is much easier to understand and is the way both the Z transform and its continuous-time counterpart the **Laplace-transform** are represented graphically.



(a)



(b)

Figure 12.1.3: Magnitude and Phase of $H(z)$.

What could the system H be doing? It is a perfect all-pass, linear-phase system. But what does this mean?

Suppose $h[n] = \delta[n - n_0]$. Then

$$\begin{aligned} H(z) &= \sum_{n=-\infty}^{\infty} h[n] z^{-n} \\ &= \sum_{n=-\infty}^{\infty} \delta[n - n_0] z^{-n} \\ &= z^{-n_0} \end{aligned}$$

Thus, $H(z) = z^{-n_0}$ is the z-transform of a system that simply delays the input by n_0. $H(z)$ is the z-transform of a unit-delay.

Now consider $x[n] = \alpha^n u[n]$

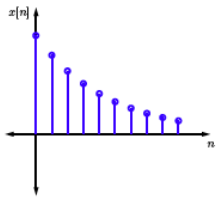


Figure 12.1.4

$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{\infty} x[n] z^{-n} = \sum_{n=0}^{\infty} \alpha^n z^{-n} \\ &= \sum_{n=0}^{\infty} \left(\frac{\alpha}{z}\right)^n \\ &= \frac{1}{1 - \frac{\alpha}{z}} \left(\text{ if } \left|\frac{\alpha}{z}\right| < 1 \right) \text{ (Geometric series)} \\ &= \frac{z}{z - \alpha} \end{aligned} \tag{12.1.1}$$

What if $|\frac{\alpha}{z}| \geq 1$? Then $\sum_{n=0}^{\infty} (\frac{\alpha}{z})^n$ does not converge! Therefore, whenever we compute a z-tranform, we must also specify the set of z's for which the z-transform exists. This is called the *region of convergence* (ROC).

Note: Using a computer to find the Z-transform

Matlab has two functions,

```
ztrans
```

and

```
iztrans
```

that are both part of the symbolic toolbox, and will find the Z and inverse Z transforms respectively. This method is generally preferred for more complicated functions. Simpler and more contrived functions are usually found easily enough by using tables.

Application to Discrete Time Filters

The z-transform might seem slightly ugly. We have to worry about the region of convergence, and stability issues, and so forth. However, in the end it is worthwhile because it proves extremely useful in analyzing digital filters with feedback. For example, consider the system illustrated below

Plots

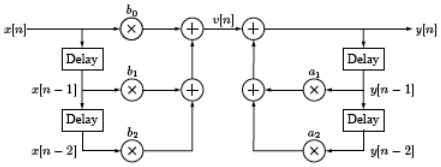


Figure 12.1.5

We can analyze this system via the equations

$$v[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

and

$$y[n] = v[n] + a_1 y[n-1] + a_2 y[n-2]$$

More generally,

$$v[n] = \sum_{k=0}^{N} b_k x[n-k]$$

and

$$y[n] = \sum_{k=1}^{M} a_k y[n-k] + v[n]$$

or equivalently,

$$\sum_{k=0}^{N} b_k x[n-k] = y[n] - \sum_{k=1}^{M} a_k y[n-k].$$

What does the z-transform of this relationship look like?

$$Z \sum_{k=0}^{M} a_k y[n-k] = Z \sum_{k=0}^{M} b_k x[n-k]$$

$$\sum_{k=0}^{M} a_k Z\{y[n-k]\} = \sum_{k=0}^{M} b_k Z\{x[n-k]\}$$

Note that

$$\begin{aligned} Z\{y[n-k]\} &= \sum_{n=-\infty}^{\infty} y[n-k] z^{-n} \\ &= \sum_{m=-\infty}^{\infty} y[m] z^{-m} z^{-k} \\ &= Y(z) z^{-k} \end{aligned}$$

Thus the relationship reduces to

$$\begin{aligned} \sum_{k=0}^{M} a_k Y(z) z^{-k} &= \sum_{k=0}^{N} b_k X(z) z^{-k} \\ Y(z) \sum_{k=0}^{M} a_k z^{-k} &= X(z) \sum_{k=0}^{N} b_k z^{-k} \\ \frac{Y(z)}{X(z)} &= \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{M} a_k z^{-k}} \end{aligned} \tag{12.1.2}$$

Hence, given a system the one above, we can easily determine the system's transfer function, and end up with a ratio of two polynomials in \(z\): a rational function. Similarly, given a rational function, it is easy to realize this function in a simple hardware architecture.

Interactive Z-Transform Demonstration

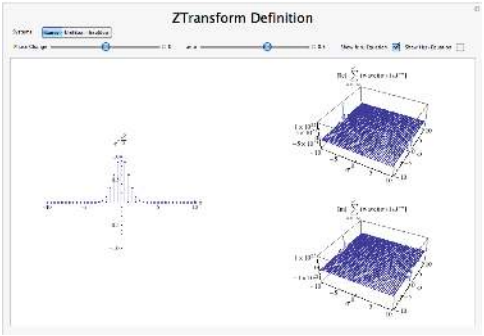


Figure 12.1.6: Interact (when online) with a Mathematica CDF demonstrating the Z Transform. To Download, right-click and save target as .cdf.

Conclusion

The z-transform proves a useful, more general form of the Discrete Time Fourier Transform. It applies equally well to describing systems as well as signals using the eigenfunction method, and proves extremely useful in digital filter design.

This page titled 12.1: Z-Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

12.2: Common Z-Transforms

Table 12.2.1: Common Continuous Time Fourier Series

Signal	Z-transform	Region of Convergence
$\delta(t)$	1	all s
$\delta(t-T)$	$e^{-(sT)}$	all s
$u(t)$	$\frac{1}{s}$	$\mathbb{R}>0$
$-u(-t)$	$\frac{1}{s}$	$\mathbb{R}<0$
$tu(t)$	$\frac{1}{s^2}$	$\mathbb{R}>0$
$t^n u(t)$	$\frac{n!}{s^{n+1}}$	$\mathbb{R}>0$
$-(t^n u(-t))$	$\frac{n!}{s^{n+1}}$	$\mathbb{R}<0$
$e^{-(\lambda t)}u(t)$	$\frac{1}{s+\lambda}$	$\mathbb{R}>-\lambda$
$-\left(e^{-(\lambda t)}\right)u(-t)$	$\frac{1}{s+\lambda}$	$\mathbb{R}<-\lambda$
$te^{-(\lambda t)}u(t)$	$\frac{1}{(s-\lambda)^2}$	$\mathbb{R}>-\lambda$
$t^n e^{-(\lambda t)}u(t)$	$\frac{n!}{(s+\lambda)^{n+1}}$	$\mathbb{R}>-\lambda$
$-\left(t^n e^{-(\lambda t)}u(-t)\right)$	$\frac{n!}{(s+\lambda)^{n+1}}$	$\mathbb{R}<-\lambda$
$\cos(bt)u(t)$	$\frac{s}{s^2+b^2}$	$\mathbb{R}>0$
$\sin(bt)u(t)$	$\frac{b}{s^2+b^2}$	$\mathbb{R}>0$
$e^{-(at)}\cos(bt)u(t)$	$\frac{s+a}{(s+a)^2+b^2}$	$\mathbb{R}>-a$
$e^{-(at)}\sin(bt)u(t)$	$\frac{b}{(s+a)^2+b^2}$	$\mathbb{R}>-a$
$\frac{d^n}{dt^n}\delta(t)$	s^n	all s

This page titled 12.2: Common Z-Transforms is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

12.3: Properties of the Z-Transform

Introduction

This module will look at some of the basic properties of the Z-Transform (Section 9.2) (DTFT).

Note

We will be discussing these properties for aperiodic, discrete-time signals but understand that very similar properties hold for continuous-time signals and periodic signals as well.

Discussion of Z-Transform Properties

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 Z-transform of a linear combination of signals then it will be the same as the linear combination of the Z-transforms of each of the individual signals. This is crucial when using a table (Section 8.3) of transforms to find the transform of a more complicated signal.

Example 12.3.1

We will begin with the following signal:

$$x[n] = af_1[n] + bf_2[n]$$

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.

$$X(z) = aF_1(z) + bF_2(z)$$

Symmetry

Symmetry is a property that can make life quite easy when solving problems involving Z-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 symmetry between the forward Z and the inverse Z transform. The only difference is the scaling by 2π and a frequency reversal.

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 pulse 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.

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 is proven below:

Example 12.3.2

We will begin by letting $x[n] = f[n-\eta]$. Now let's take the z-transform with the previous expression substituted in for $x[n]$.

$$X(z) = \sum_{n=-\infty}^{\infty} f[n-\eta]z^{-n}$$

Now let's make a simple change of variables, where $\sigma = n - \eta$. 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.

$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{\infty} f[\sigma]z^{-(\sigma+\eta)} \\ &= z^{-\eta}\sum_{\sigma=-\infty}^{\infty} f[\sigma]z^{-\sigma} \\ &= z^{-\eta}F(z) \end{aligned} \tag{12.3.1}$$

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 discrete-time convolution (Section 4.3) module for a more in depth explanation and derivation.

$$\begin{aligned} y[n] &= (f_1[n], f_2[n]) \\ &= \sum_{\eta=-\infty}^{\infty} f_1[\eta]f_2[n-\eta] \end{aligned} \tag{12.3.2}$$

Time Differentiation

Since discrete LTI (Section 2.1) systems can be represented in terms of difference equations, it is apparent with this property that converting to the frequency domain may allow us to convert these complicated difference equations to simpler equations involving multiplication and addition.

Parseval's Relation

$$\sum_{n=-\infty}^{\infty} x[n]x*[n] = \int_{-\pi}^{\pi} F(z)F*(z)dz$$

Parseval's relation tells us that the energy of a signal is equal to the energy of its Fourier transform.

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; however, here we would use the inverse Fourier transform in place of the Fourier transform. Since we went through the steps in the previous, time-shift proof, below we will just show the initial and final step to this proof:

$$z(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega-\phi)e^{j\omega t}d\omega$$

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:

$$z(t) = f(t)e^{j\phi t}$$

Properties Demonstration

An interactive example demonstration of the properties is included below:

Figure 12.3.1: Interactive Signal Processing Laboratory Virtual Instrument created using NI's Labview.

Summary Table

Table 12.3.1: Properties of the Z-Transform

Property	Signal	Z-Transform	Region of Convergence
Linearity	$\alpha x_1(n) + \beta x_2(n)$	$\alpha X_1(z) + \beta X_2(z)$	At least $\mathrm{ROC}_1 \cap \mathrm{ROC}_2$
Time shifing	$x(n-k)$	$z^{-k}X(z)$	ROC
Time scaling	$x(n/k)$	$X(z^k)$	$\mathrm{ROC}^{1/k}$
Z-domain scaling	$a^n x(n)$	$X(z/a)$	$\lvert a \rvert$ ROC
Conjugation	$\overline{x(n)}$	$\overline{X}(\overline{z})$	ROC
Convolution	$x_1(n) * x_2(n)$	$X_1(z)X_2(z)$	At least $\mathrm{ROC}_1 \cap \mathrm{ROC}_2$
Differentiation in z-Domain	$[nx[n]]$	$-\frac{d}{dz}X(z)$	ROC = all $\mathbb{R}$
Parseval's Theorem	$\sum_{n=-\infty}^{\infty} x[n]\mathrm{x} * [n]$	$\int_{-\pi}^{\pi} F(z)\mathbf{F} * (z)dz$	ROC

This page titled 12.3: Properties of the Z-Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

12.4: Inverse Z-Transform

Introduction

When using the z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

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
2. Partial-Fraction Expansion
3. Power Series Expansion
4. Contour Integration

Inspection Method

This "method" is to basically become familiar with the z-transform pair tables and then "reverse engineer".

Example 12.4.1

When given

$$X(z) = \frac{z}{z - \alpha}$$

with an ROC (Section 12.6) of

$$|z| > \alpha$$

we could determine "by inspection" that

$$x[n] = \alpha^n u[n]$$

Partial-Fraction Expansion Method

When dealing with **linear time-invariant systems** the z-transform is often of the form

$$\begin{aligned} X(z) &= \frac{B(z)}{A(z)} \\ &= \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \end{aligned} \tag{12.4.1}$$

This can also expressed as

$$X(z) = \frac{a_0}{b_0} \frac{\prod_{k=1}^{M} 1 - c_k z^{-1}}{\prod_{k=1}^{N} 1 - d_k z^{-1}}$$

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

$$X(z) = \sum_{k=1}^{N} \frac{A_k}{1 - d_k z^{-1}}$$

This form allows for easy inversions of each term of the sum using the inspection method and the transform table. If the numerator is a polynomial, however, then it becomes necessary to use partial-fraction expansion to put $X(z)$ in the above form. If $M \geq N$ then $X(z)$ can be expressed as

$$X(z) = \sum_{r=0}^{M-N} B_r z^{-r} + \frac{\sum_{k=0}^{N-1} b'_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}$$

Example 12.4.2

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 $|z| > 2$,

$$x[n] = \frac{1}{2}\delta[n] + \frac{9}{2}2^n u[n] - 4u[n]$$

Demonstration of Partial Fraction Expansion

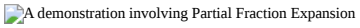


Figure 12.4.1: Interactive experiment illustrating how the Partial Fraction Expansion method is used to solve a variety of numerator and denominator problems. (To view and interact with the simulation, download the free Mathematica player at www.wolfram.com/products/player/download.cgi)

Khan Lecture on Partial Fraction Expansion

Figure 12.4.2: video from Khan Academy

Power Series Expansion Method

When the z-transform is defined as a power series in the form

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

then each term of the sequence $x[n]$ can be determined by looking at the coefficients of the respective power of z^{-n}.

Example 12.4.3

Now look at the z-transform of a **finite-length sequence**.

$$\begin{aligned} X(z) &= z^2 \left(1 + 2z^{-1}\right) \left(1 - \frac{1}{2} z^{-1}\right) \left(1 + z^{-1}\right) \\ &= z^2 + \frac{5}{2} z + \frac{1}{2} + -z^{-1} \end{aligned}$$

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 12.4.4

Suppose

$$X(z) = \log_n \left(1 + \alpha z^{-1}\right)$$

Noting that

$$\log_n(1+x) = \sum_{n=1}^{\infty} \frac{-1^{n+1}x^n}{n}$$

Then

$$X(z) = \sum_{n=1}^{\infty} \frac{-1^{n+1}\alpha^n z^{-n}}{n}$$

Therefore

$$X(z) = \begin{cases} \frac{-1^{n+1}\alpha^n}{n} \text{ if } n \geq 1 \\ 0 \text{ if } n \leq 0 \end{cases}$$

Contour Integration Method

Without going in to much detail

$$x[n] = \frac{1}{2\pi j} \oint_r X(z) z^{n-1} \mathrm{d}z$$

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.

Demonstration of Contour Integration

A demonstration involving Contour Integration

Figure 12.4.3: Interactive experiment illustrating how the contour integral is applied on a simple example. For a more in-depth discussion of this method, some background in complex analysis is required. (To view and interact with the simulation, download the free Mathematica player at www.wolfram.com/products/player/download.cgi)

Conclusion

The Inverse Z-transform is very useful to know for the purposes of designing a filter, and there are many ways in which to calculate it, drawing from many disparate areas of mathematics. All nevertheless assist the user in reaching the desired time-domain signal that can then be synthesized in hardware(or software) for implementation in a real-world filter.

This page titled 12.4: Inverse Z-Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

12.5: Poles and Zeros in the Z-Plane

Introduction to Poles and Zeros of the Z-Transform

It is quite difficult to qualitatively analyze the Laplace transform (Section 11.1) and Z-transform, 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's poles and zeros to try to gain a qualitative idea of what a system does.

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 12.7):

$$X(z) = \frac{P(z)}{Q(z)}$$

The two polynomials, $P(z)$ and $Q(z)$, allow us to find the poles and zeros of the Z-Transform.

Definition: 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: 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 12.5.1

Below is a simple transfer function with the poles and zeros shown below it.

$$H(z) = \frac{z+1}{\left(z-\frac{1}{2}\right)\left(z+\frac{3}{4}\right)}$$

The zeros are: -1

The poles are: $\left\{\frac{1}{2}, -\frac{3}{4}\right\}$

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.

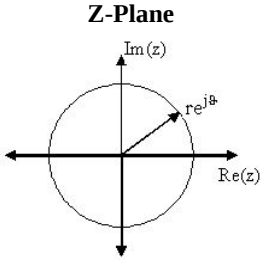


Figure 12.5.1

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 12.5.2: Simple Pole/Zero Plot

$$H(z) = \frac{z}{\left(z - \frac{1}{2}\right)\left(z + \frac{3}{4}\right)}$$

The zeros are: $\{0\}$

The poles are: $\left\{\frac{1}{2}, -\frac{3}{4}\right\}$

Pole/Zero Plot

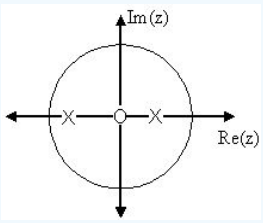


Figure 12.5.2: Using the zeros and poles found from the transfer function, the one zero is mapped to zero and the two poles are placed at $\frac{1}{2}$ and $-\frac{3}{4}$

Example 12.5.3: Complex Pole/Zero Plot

$$H(z) = \frac{(z-j)(z+j)}{\left(z - \left(\frac{1}{2} - \frac{1}{2}j\right)\right)\left(z - \frac{1}{2} + \frac{1}{2}j\right)}$$

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

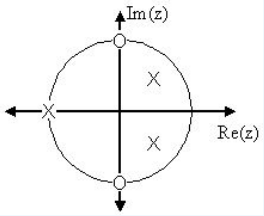


Figure 12.5.3: Using the zeros and poles found from the transfer function, the zeros are mapped to $\pm(j)$, and the poles are placed at -1, $\frac{1}{2} + \frac{1}{2}j$ and $\frac{1}{2} - \frac{1}{2}j$

Example 12.5.4: 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.

Note: 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.

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');
```

Interactive Demonstration of Poles and Zeros

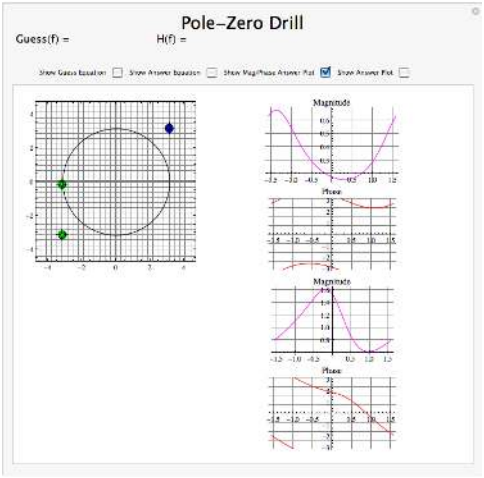


Figure 12.5.4: Interact (when online) with a Mathematica CDF demonstrating Pole/Zero Plots. To Download, right-click and save target as .cdf.

Applications for pole-zero plots

Stability and Control theory

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 3.6).

Pole/Zero Plots and the 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 12.5.2) 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 12.5.5

$$H(z) = \frac{z}{\left(z - \frac{1}{2}\right)\left(z + \frac{3}{4}\right)}$$

Region of Convergence for the Pole/Zero Plot

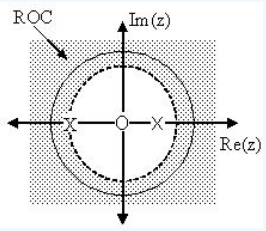


Figure 12.5.5: The shaded area represents the chosen ROC for the transfer function.

Frequency Response and Pole/Zero Plots

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.

This page titled 12.5: Poles and Zeros in the Z-Plane is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

12.6: Region of Convergence for the Z-Transform

Introduction

With the z-transform, the s-plane represents a set of signals (complex exponentials (Section 1.8)). 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 discrete-time, LTI system.

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 exists. The **z-transform** of a sequence is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

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,

$$\sum_{n=-\infty}^{\infty} |x[n]z^{-n}| < \infty$$

must be satisfied for convergence.

Properties of the Region of Convergencec

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 $\mathrm{x[n]}$ is a finite-duration sequence, then the ROC is the entire z-plane, except possibly $\mathrm{z} = 0$ or $|\mathrm{z}| = \infty$.** 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 $n_2 > 0$ 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 $x[n] = c\delta[n]$.

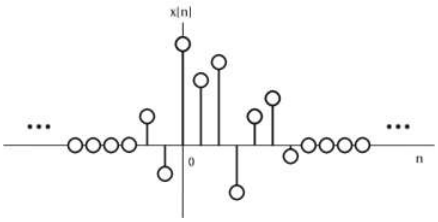


Figure 12.6.1: An example of a finite duration sequence.

The next properties apply to infinite duration sequences. As noted above, the z-transform converges when $|X(z)| < \infty$. So we can write

$$|X(z)| = \left|\sum_{n=-\infty}^{\infty} x[n]z^{-n}\right| \leq \sum_{n=-\infty}^{\infty} |x[n]z^{-n}| = \sum_{n=-\infty}^{\infty} |x[n]|(|z|)^{-n}$$

We can then split the infinite sum into positive-time and negative-time portions. So

$$|X(z)| \leq N(z) + P(z)$$

where

$$N(z) = \sum_{n=-\infty}^{-1} |x[n]|(|z|)^{-n}$$

and

$$P(z) = \sum_{n=0}^{\infty} |x[n]|(|z|)^{-n}$$

In order for $|X(z)|$ to be finite, $|x[n]|$ must be bounded. Let us then set

$$|x(n)| \leq C_1 r_1^n$$

for

$$n < 0$$

and

$$|x(n)| \leq C_2 r_2^n$$

for

$$n \geq 0$$

From this some further properties can be derived:

- **If $\mathbf{x[n]}$ is a right-sided sequence, then the ROC extends outward from the outermost pole in $\mathbf{X(z)}$.** 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

$$P(z) \leq C_2 \sum_{n=0}^{\infty} r_2^n (|z|)^{-n} = C_2 \sum_{n=0}^{\infty} \left(\frac{r_2}{|z|}\right)^n$$

  Thus in order for this sum to converge, $|z| > r_2$, and therefore the ROC of a right-sided sequence is of the form $|z| > r_2$.

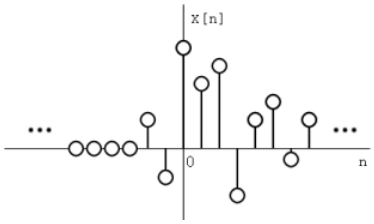


Figure 12.6.2: A right-sided sequence.

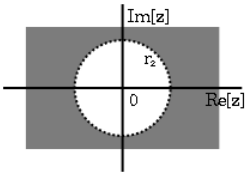


Figure 12.6.3: The ROC of a right-sided sequence.

- **If $\mathbf{x[n]}$ is a left-sided sequence, then the ROC extends inward from the innermost pole in $\mathbf{X(z)}$.** A **left-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

$$N(z) \leq C_1 \sum_{n=-\infty}^{-1} r_1^n (|z|)^{-n} = C_1 \sum_{n=-\infty}^{-1} \left(\frac{r_1}{|z|}\right)^n = C_1 \sum_{k=1}^{\infty} \left(\frac{|z|}{r_1}\right)^k$$

  Thus in order for this sum to converge, $|z| < r_1$, and therefore the ROC of a left-sided sequence is of the form $|z| < r_1$.

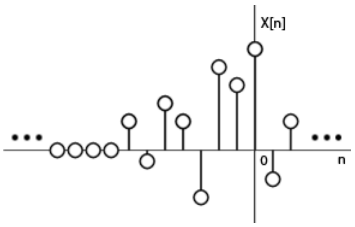


Figure 12.6.4: A left-sided sequence.

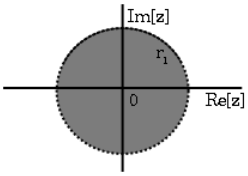


Figure 12.6.5: The ROC of a left-sided sequence.

- **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 $-r_2 < |z| < r_2$.

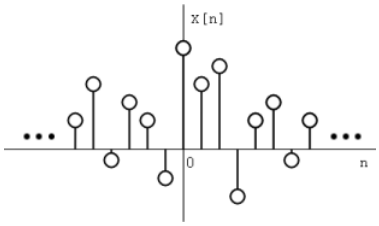


Figure 12.6.6: A two-sided sequence.

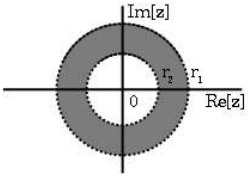


Figure 12.6.7: The ROC of a two-sided sequence.

Examples

Example 12.6.1

Let's take

$$x_1[n] = \left(\frac{1}{2}\right)^n u[n] + \left(\frac{1}{4}\right)^n u[n]$$

The z-transform of $\left(\frac{1}{2}\right)^n u[n]$ is $\frac{z}{z-\frac{1}{2}}$ with an ROC at $|z| > \frac{1}{2}$.

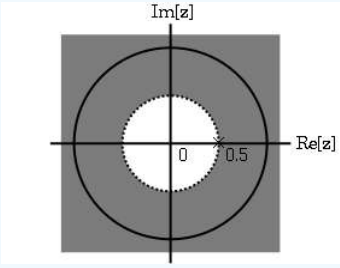


Figure 12.6.8: The ROC of $\left(\frac{1}{2}\right)^n u[n]$

The z-transform of $\left(\frac{-1}{4}\right)^n u[n]$ is $\frac{z}{z+\frac{1}{4}}$ with an ROC at $|z| > \frac{-1}{4}$.

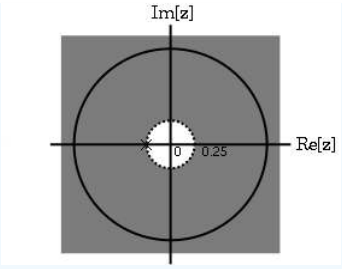


Figure 12.6.9: The ROC of $\left(\frac{-1}{4}\right)^n u[n]$

Due to linearity,

$$\begin{aligned} X_1[z] &= \frac{z}{z-\frac{1}{2}} + \frac{z}{z+\frac{1}{4}} \\ &= \frac{2z\left(z-\frac{1}{8}\right)}{\left(z-\frac{1}{2}\right)\left(z+\frac{1}{4}\right)} \end{aligned} \tag{12.6.1}$$

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 obove properties, the ROC is $|z| > \frac{1}{2}$.

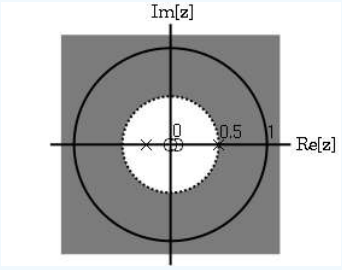


Figure 12.6.10: The ROC of $x_1[n] = \left(\frac{1}{2}\right)^n u[n] + \left(\frac{-1}{4}\right)^n u[n]$

Example 12.6.2

Now take

$$x_2[n] = \left(\frac{-1}{4}\right)^n u[n] - \left(\frac{1}{2}\right)^n u[(-n)-1]$$

The z-transform and ROC of $\left(\frac{-1}{4}\right)^n u[n]$ was shown in the example above. The z-transorm of $\left(-\left(\frac{1}{2}\right)^n\right) u[(-n)-1]$ is $\frac{z}{z-\frac{1}{2}}$ with an ROC at $|z| > \frac{1}{2}$.

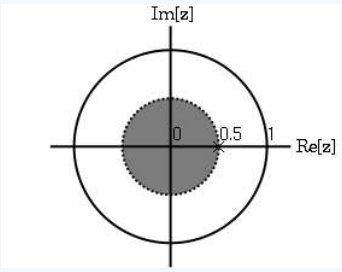


Figure 12.6.11: The ROC of $\left(-\left(\frac{1}{2}\right)^n\right) u[(-n)-1]$

Once again, by linearity,

$$\begin{aligned} X_2[z] &= \frac{z}{z+\frac{1}{4}} + \frac{z}{z-\frac{1}{2}} \\ &= \frac{z\left(2z-\frac{1}{8}\right)}{\left(z+\frac{1}{4}\right)\left(z-\frac{1}{2}\right)} \end{aligned} \tag{12.6.2}$$

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 ths case though, the ROC is $|z| < \frac{1}{2}$.

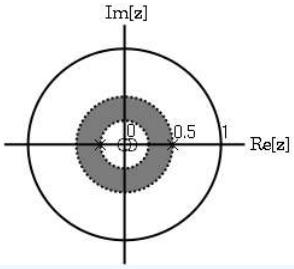


Figure 12.6.12: The ROC of $x_2[n] = \left(\frac{-1}{4}\right)^n u[n] - \left(\frac{1}{2}\right)^n u[(-n) - 1]$.

Graphical Understanding of ROC

Using the demonstration, learn about the region of convergence for the Laplace Transform.

Conclusion

Clearly, in order to craft a system that is actually useful by virtue of being causal and BIBO stable, we must ensure that it is within the Region of Convergence, which can be ascertained by looking at the pole zero plot. The Region of Convergence is the area in the pole/zero plot of the transfer function in which the function exists. For purposes of useful filter design, we prefer to work with rational functions, which can be described by two polynomials, one each for determining the poles and the zeros, respectively.

This page titled 12.6: Region of Convergence for the Z-Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

12.7: Rational Functions and the Z-Transform

Introduction

When dealing with operations on polynomials, the term **rational function** is a simple way to describe a particular relationship between two polynomials.

Definition: Rational Function

For any two polynomials, A and B, their quotient is called a rational function.

Example 12.7.1

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.

$$f(x) = \frac{x^2 - 4}{2x^2 + x - 3} \tag{12.7.1}$$

If you have begun to study the Z-transform, 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.

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 subsection to see how rational functions are useful when dealing with the z-transform.

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, Equation 12.7.1.

$$f(x) = \frac{(x+2)(x-2)}{(2x+3)(x-1)} \tag{12.7.2}$$

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.

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.

Example 12.7.2

Continuing to look at our rational function above, Equation 12.7.1, 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.

Domain

Using the roots that we found above, the **domain** of the rational function can be easily defined.

Definition: Domain

The group, or set, of values that are defined by a given function.

Example 12.7.3

Using the rational function above, Equation 12.7.1, the domain can be defined as any real number x where x does not equal 1 or negative 3. Written out mathematical, we get the following:

$$\{x \in \mathbb{R} \mid (x \neq -3) \text{ and } (x \neq 1)\} \tag{12.7.3}$$

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.

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:

$$X(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}} \tag{12.7.4}$$

If you have already looked at the module about Understanding Pole/Zero Plots and the Z-transform (Section 12.5), you should see how the roots of the rational function play an important role in understanding the z-transform. The equation above, Equation 12.7.4, can be expressed in factored form just as was done for the simple rational function above, see Equation 12.7.2. 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

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 12.5), or to analyze and design a digital filter through Filter Design from the Z-Transform (Section 12.9).

This page titled 12.7: Rational Functions and the Z-Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

12.8: Difference Equations

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: 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 12.8.1

$$y[n] + 7y[n-1] + 2y[n-2] = x[n] - 4x[n-1]$$

General Formulas for 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.

Difference Equation

The general form of a linear, constant-coefficient difference equation (LCCDE), is shown below:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \tag{12.8.1}$$

We can also write the general form to easily express a recursive output, which looks like this:

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] \tag{12.8.2}$$

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.

Conversion to Z-Transform

Using the above formula, Equation 12.8.2, 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 Transform of all the terms in Equation 12.8.2. 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: $a_0 = 1$.

$$Y(z) = -\sum_{k=1}^{N} a_k Y(z) z^{-k} + \sum_{k=0}^{M} b_k X(z) z^{-k}$$

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} \\ &= \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \end{aligned} \tag{12.8.3}$$

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}.

$$H(w) = H(z)|_{z,z=e^{jw}} \tag{12.8.4}$$

$$= \frac{\sum_{k=0}^{M} b_k e^{-(jwk)}}{\sum_{k=0}^{N} a_k e^{-(jwk)}} \tag{12.8.5}$$

Once you understand the derivation of this formula, look at the module concerning Filter Design from the Z-Transform (Section 12.9) for a look into how all of these ideas of the Z-transform, Difference Equation, and Pole/Zero Plots (Section 12.5) play a role in filter design.

Example

Example 12.8.2: 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.

$$H(z) = \frac{(z+1)^2}{\left(z-\frac{1}{2}\right)\left(z+\frac{3}{4}\right)}$$

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.

$$\begin{aligned} H(z) &= \frac{(z+1)(z+1)}{\left(z-\frac{1}{2}\right)\left(z+\frac{3}{4}\right)} \\ &= \frac{z^2+2z+1}{z^2+2z+1-\frac{3}{8}} \\ &= \frac{1+2z^{-1}+z^{-2}}{1+\frac{1}{4}z^{-1}-\frac{3}{8}z^{-2}} \end{aligned} \tag{12.8.6}$$

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, Equation 12.8.2. Using these coefficients and the above form of the transfer function, we can easily write the difference equation:

$$x[n] + 2x[n-1] + x[n-2] = y[n] + \frac{1}{4}y[n-1] - \frac{3}{8}y[n-2]$$

In our final step, we can rewrite the difference equation in its more common form showing the recursive nature of the system.

$$y[n] = x[n] + 2x[n-1] + x[n-2] + \frac{-1}{4}y[n-1] + \frac{3}{8}y[n-2]$$

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.

Direct Method

The final solution to the output based on the direct method is the sum of two parts, expressed in the following equation:

$$y(n) = y_h(n) + y_p(n)$$

The first part, $y_h(n)$, is referred to as the **homogeneous solution** and the second part, $y_h(n)$, 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.

Homogeneous Solution

We begin by assuming that the input is zero, $x(n) = 0$. Now we simply need to solve the homogeneous difference equation:

$$\sum_{k=0}^{N} a_k y[n-k] = 0$$

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:

$$\sum_{k=0}^{N} a_k \lambda^{n-k} = 0$$

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:

$$y_h(n) = C_1 (\lambda_1)^n + C_2 (\lambda_2)^n + \cdots + C_N (\lambda_N)^n$$

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:

$$y_h(n) = C_1 (\lambda_1)^n + C_1 n (\lambda_1)^n + C_1 n^2 (\lambda_1)^n + \cdots + C_1 n^{K-1} (\lambda_1)^n + C_2 (\lambda_2)^n + \cdots + C_N (\lambda_N)^n$$

Particular Solution

The particular solution, $y_p(n)$, will be any solution that will solve the general difference equation:

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

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.

Indirect Method

The indirect method utilizes the relationship between the difference equation and z-transform, discussed earlier, to find a solution. The basic idea is to convert the difference equation into a z-transform, as described above, to get the resulting output, $Y(z)$. Then by inverse transforming this and using partial-fraction expansion, we can arrive at the solution.

$$Z\{y(n+1) - y(n)\} = zY(z) - y(0)$$

This can be interatively extended to an arbitrary order derivative as in Equation 12.8.7.

$$Z\left\{-\sum_{m=0}^{N-1} y(n-m)\right\} = z^n Y(z) - \sum_{m=0}^{N-1} z^{n-m-1} y^{(m)}(0) \tag{12.8.7}$$

Now, the Laplace transform of each side of the differential equation can be taken

$$Z\left\{\sum_{k=0}^{N} a_k \left[y(n-m+1) - \sum_{m=0}^{N-1} y(n-m)y(n)\right] = Z\{x(n)\}\right\}$$

which by linearity results in

$$\sum_{k=0}^{N} a_k Z\left\{y(n-m+1) - \sum_{m=0}^{N-1} y(n-m)y(n)\right\} = Z\{x(n)\}$$

and by differentiation properties in

$$\sum_{k=0}^{N} a_k \left(z^k Z\{y(n)\} - \sum_{m=0}^{N-1} z^{k-m-1} y^{(m)}(0)\right) = Z\{x(n)\}.$$

Rearranging terms to isolate the Laplace transform of the output,

$$Z\{y(n)\} = \frac{Z\{x(n)\} + \sum_{k=0}^{N}\sum_{m=0}^{k-1} a_k z^{k-m-1} y^{(m)}(0)}{\sum_{k=0}^{N} a_k z^k}.$$

Thus, it is found that

$$Y(z) = \frac{X(z) + \sum_{k=0}^{N}\sum_{m=0}^{k-1} a_k z^{k-m-1} y^{(m)}(0)}{\sum_{k=0}^{N} a_k z^k}. \tag{12.8.8}$$

In order to find the output, it only remains to find the Laplace transform $X(z)$ of the input, substitute the initial conditions, and compute the inverse Z-transform of the result. Partial fraction expansions are often required for this last step. This may sound daunting while looking at Equation 12.8.8, but it is often easy in practice, especially for low order difference equations. Equation 12.8.8 can also be used to determine the transfer function and frequency response.

As an example, consider the difference equation

$$y[n-2] + 4y[n-1] + 3y[n] = \cos(n)$$

with the initial conditions $y'(0) = 1$ and $y(0) = 0$ Using the method described above, the Z transform of the solution $y[n]$ is given by

$$Y[z] = \frac{z}{[z^2+1]\,[z+1][z+3]} + \frac{1}{[z+1][z+3]}.$$

Performing a partial fraction decomposition, this also equals

$$Y[z] = .25\frac{1}{z+1} - .35\frac{1}{z+3} + .1\frac{z}{z^2+1} + .2\frac{1}{z^2+1}.$$

Computing the inverse Laplace transform,

$$y(n) = \left(.25z^{-n} - .35z^{-3n} + .1\cos(n) + .2\sin(n)\right)u(n).$$

One can check that this satisfies that this satisfies both the differential equation and the initial conditions.

This page titled 12.8: Difference Equations is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

12.9: Discrete Time Filter Design

Estimating Frequency Response from Z-Plane

One of the primary motivating factors for utilizing the z-transform and analyzing the pole/zero plots is due to their relationship to the frequency response of a discrete-time 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:

$$\begin{aligned} H(w) &= H(z)|_{z,z=e^{jw}} \\ &= \frac{\sum_{k=0}^{M} b_k e^{-(jwk)}}{\sum_{k=0}^{N} a_k e^{-(jwk)}} \end{aligned} \tag{12.9.1}$$

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:

$$H(w) = \left|\frac{b_0}{a_0}\right| \frac{\prod_{k=1}^{M} |e^{jw} - c_k|}{\prod_{k=1}^{N} |e^{jw} - d_k|} \tag{12.9.2}$$

From Equation 12.9.2 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 ww goes from zero to pi. These characteristics allow us to interpret $|H(w)|$ as follows:

$$|H(w)| = \left|\frac{b_0}{a_0}\right| \frac{\prod \text{ "distances from zeros"}}{\prod \text{ "distances from poles"}}$$

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 12.5) module.

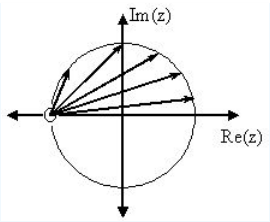
Example 12.9.1

In this first example we will take a look at the very simple z-transform shown below:

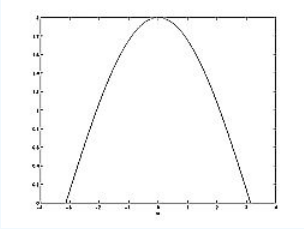
$$H(z) = z + 1 = 1 + z^{-1}$$

$$H(w) = 1 + e^{-(jw)}$$

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 Figure 12.9.1 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 Equation 12.9.2 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 ww 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 Equation 12.9.2.



(a) Pole/Zero Plot



(b) Frequency Response: $|H(w)|$

Figure 12.9.1: 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.

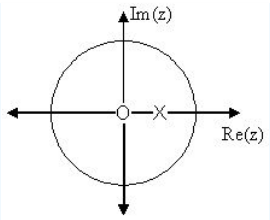
Example 12.9.2

For this example, a more complex transfer function is analyzed in order to represent the system's frequency response.

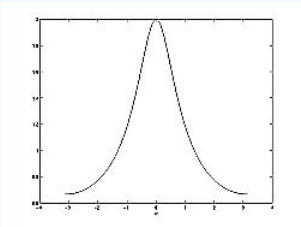
$$H(z) = \frac{z}{z - \frac{1}{2}} = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

$$H(w) = \frac{1}{1 - \frac{1}{2}e^{-(jw)}}$$

Below we can see the two figures described by the above equations. The Figure $12.9.2(a)$ represents the basic pole/zero plot of the z-transform, $H(w)$. Figure $12.9.2(b)$ 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 ww that the pole is closest to the unit circle. The ratio from Equation 12.9.2 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 12.9.2: 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.

Figure 12.9.3: Digital filter design LabVIEW virtual instrument by NI from http://cnx.org/content/m13115/latest/.

Conclusion

In conclusion, using the distances from the unit circle to the poles and zeros, we can plot the frequency response of the system. As ww 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.

This page titled 12.9: Discrete Time Filter Design is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

13: Capstone Signal Processing Topics

13.1: DFT- Fast Fourier Transform

13.2: The Fast Fourier Transform (FFT)

13.3: Deriving the Fast Fourier Transform

13.4: Matched Filter Detector

This page titled 13: Capstone Signal Processing Topics is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

13.1: DFT- Fast Fourier Transform

We now have a way of computing the spectrum for an arbitrary signal: The Discrete Fourier Transform (DFT) 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 13.1.1

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 $\left(k=\left[\frac{N}{2}+1,\ldots,N+1\right]\right.$ in the DFT) 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?

Answer

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 $O(KN)$.

This page titled 13.1: DFT- Fast Fourier Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

13.2: The Fast Fourier Transform (FFT)

Introduction

The Fast Fourier Transform (FFT) is an efficient O(NlogN) algorithm for calculating DFTs The FFT exploits symmetries in the W matrix to take a "divide and conquer" approach. We will first discuss deriving the actual FFT algorithm, some of its implications for the DFT, and a speed comparison to drive home the importance of this powerful algorithm.

Deriving the FFT

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.

$$\begin{aligned} S(k) &= s(0) + s(2)e^{(-j)\frac{2\pi 2k}{N}} + \ldots + s(N-2)e^{(-j)\frac{2\pi(N-2)k}{N}} + s(1)e^{(-j)\frac{2\pi k}{N}} \\ &\quad + s(3)e^{(-j)\frac{2\pi\times(2+1)k}{N}} + \ldots + s(N-1)e^{(-j)\frac{2\pi(N-2+1)k}{N}} \\ &= s(0) + s(2)e^{(-j)\frac{2\pi k}{\frac{N}{2}}} + \ldots + s(N-2)e^{(-j)\frac{2\pi\left(\frac{N}{2}-1\right)k}{\frac{N}{2}}} \\ &\quad + \left(s(1) + s(3)e^{(-j)\frac{2\pi k}{\frac{N}{2}}} + \cdots + s(N-1)e^{(-j)\frac{2\pi\left(\frac{N}{2}-1\right)k}{\frac{N}{2}}} \right) e^{\frac{-(j2\pi k)}{N}} \end{aligned} \tag{13.2.1}$$

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 $k \in \{0, \ldots, N-1\}$. 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 13.2.1 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 13.2.1). 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 $O(N \log N)$.

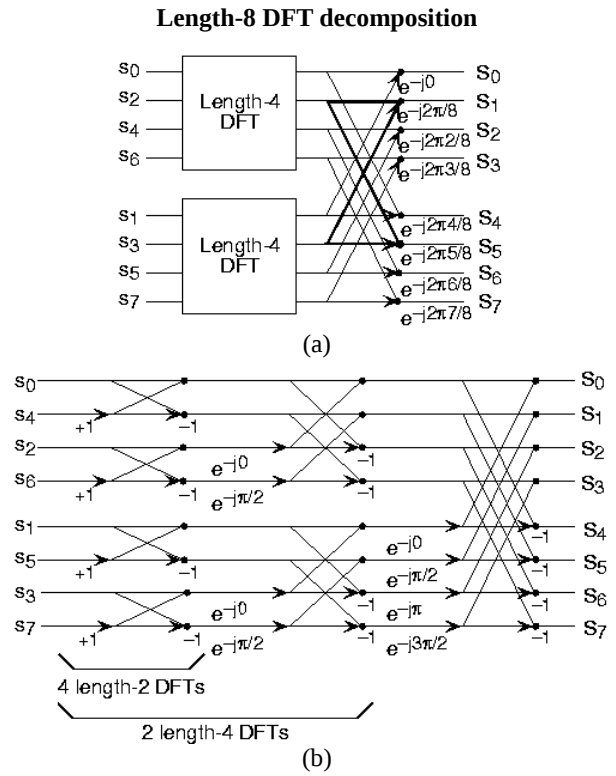


Figure 13.2.1: The initial decomposition of a length-8 DFT into the terms using even- and odd-indexed inputs marks the first phase of developing the FFT algorithm. When these half-length transforms are successively decomposed, we are left with the diagram shown in the bottom panel that depicts the length-8 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 13.2.1, we first decompose the DFT into two length-4 DFTs, with the outputs added and subtracted together in pairs. Considering Figure 13.2.1 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 13.2.2).

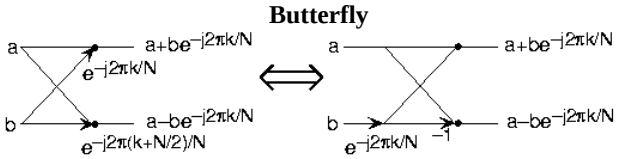


Figure 13.2.2: The basic computational element of the fast Fourier transform is the butterfly. It takes two complex numbers, represented by **a** and **b**, and forms the quantities shown. Each butterfly requires one complex multiplication and two complex additions.

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 13.2.1). 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 13.2.1

Note that the ordering of the input sequence in the two parts of Figure 13.2.1 aren't quite the same. Why not? How is the ordering determined?

Answer

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.

FFT and the DFT

We now have a way of computing the spectrum for an arbitrary signal: The Discrete Fourier Transform (DFT) 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 13.2.2

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 ($k=\left[\frac{N}{2}+1,\ldots,N+1\right]$ in the DFT) 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?

Answer

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 $O(KN)$.

Speed Comparison

How much better is $O(N\log N)$ than $O(N^2)$?

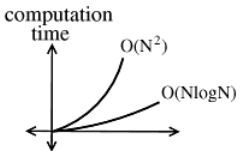


Figure 13.2.3: This figure shows how much slower the computation time of an $O(N\log N)$ process grows.

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 million=10^6.

An $O(N^2)$ algorithm takes 10^{12} Flors $\rightarrow$ 10^6 seconds $\simeq$ 11.5 days.

An $O(N \log N)$ algorithm takes 6×10^6 Flors $\rightarrow$ 6 seconds.

Note

N = 1 million is not unreasonable.

Example 13.2.1

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 $f[n] \circledast h[n]$ directly: $O(N^2)$ operations.

taking FFTs -- $O(N \log N)$

multiplying FFTs -- $O(N)$

inverse FFTs -- $O(N \log N)$.

the total complexity is $O(N \log N)$.

Conclusion

Other "fast" algorithms have been discovered, most 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. It is even well established that the FFT, alongside the digital computer, were almost completely responsible for the "explosion" of DSP in the 60's.

This page titled 13.2: The Fast Fourier Transform (FFT) is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

13.3: Deriving the Fast Fourier Transform

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.

$$\begin{aligned} S(k) &= s(0) + s(2)e^{(-j)\frac{2\pi 2k}{N}} + \ldots + s(N-2)e^{(-j)\frac{2\pi(N-2)k}{N}} + s(1)e^{(-j)\frac{2\pi k}{N}} \\ &\quad + s(3)e^{(-j)\frac{2\pi\times(2+1)k}{N}} + \ldots + s(N-1)e^{(-j)\frac{2\pi(N-2+1)k}{N}} \\ &= s(0) + s(2)e^{(-j)\frac{2\pi k}{\frac{N}{2}}} + \ldots + s(N-2)e^{(-j)\frac{2\pi\left(\frac{N}{2}-1\right)k}{\frac{N}{2}}} \\ &\quad + \left(s(1) + s(3)e^{(-j)\frac{2\pi k}{\frac{N}{2}}} + \cdots + s(N-1)e^{(-j)\frac{2\pi\left(\frac{N}{2}-1\right)k}{\frac{N}{2}}} \right) e^{\frac{-(j2\pi k)}{N}} \end{aligned} \tag{13.3.1}$$

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 $k \in \{0, \ldots, N-1\}$. 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 13.3.1 illustrates this decomposition. As it stands, we now compute two length- $\frac{N}{2}$ transforms (complexity $2O(\frac{N^2}{4})$), 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 13.3.1). 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 $O(N \log N)$.

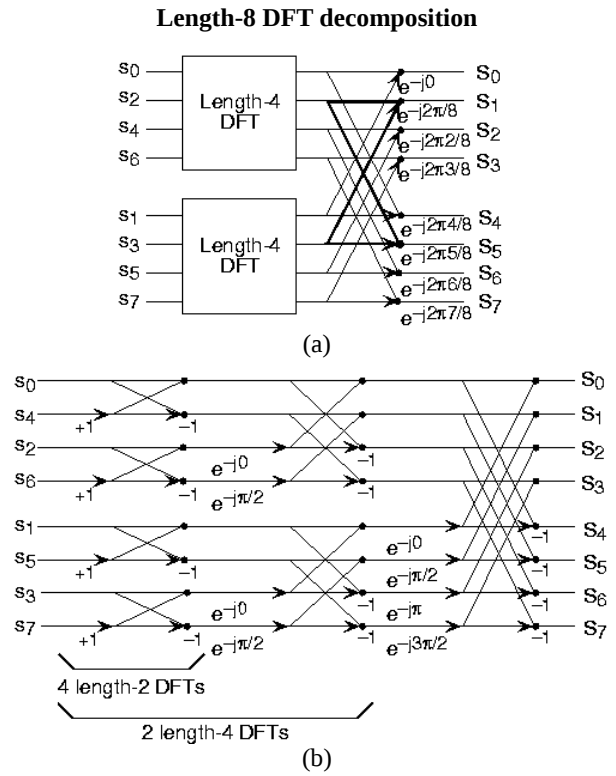


Figure 13.3.1: The initial decomposition of a length-8 DFT into the terms using even- and odd-indexed inputs marks the first phase of developing the FFT algorithm. When these half-length transforms are successively decomposed, we are left with the diagram shown in the bottom panel that depicts the length-8 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 13.3.1, we first decompose the DFT into two length-4 DFTs, with the outputs added and subtracted together in pairs. Considering Figure 13.3.1 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 13.3.2).

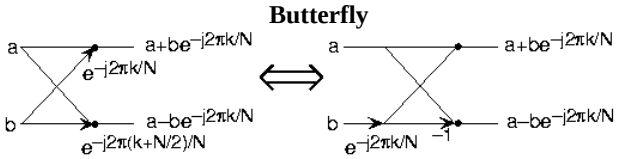


Figure 13.3.2: The basic computational element of the fast Fourier transform is the butterfly. It takes two complex numbers, represented by **a** and **b**, and forms the quantities shown. Each butterfly requires one complex multiplication and two complex additions.

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 13.3.1). 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 13.3.1

Note that the ordering of the input sequence in the two parts of Figure 13.3.1 aren't quite the same. Why not? How is the ordering determined?

Answer

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.

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.

This page titled 13.3: Deriving the Fast Fourier Transform is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

13.4: Matched Filter Detector

Introduction

A great many applications in signal processing, image processing, and beyond involve determining the presence and location of a target signal within some other signal. A radar system, for example, searches for copies of a transmitted radar pulse in order to determine the presence of and distance to reflective objects such as buildings or aircraft. A communication system searches for copies of waveforms representing digital 0s and 1s in order to receive a message.

Two key mathematical tools that contribute to these applications are inner products and the Cauchy-Schwarz inequality. As is shown in the module on the Cauchy-Schwarz inequality, the expression $\left|\left\langle \frac{x}{||x||}, \frac{y}{\|y\|} \right\rangle\right|$ attains its upper bound, which is 1, when $y = ax$ for some scalar a in a real or complex field. The lower bound, which is 0, is attained when x and y are orthogonal. In informal intuition, this means that the expression is maximized when the vectors x and y have the same shape or pattern and minimized when x and y are very different. A pair of vectors with similar but unequal shapes or patterns will produce relatively large value of the expression less than 1, and a pair of vectors with very different but not orthogonal shapes or patterns will produce relatively small values of the expression greater than 0. Thus, the above expression carries with it a notion of the degree to which two signals are "alike", the magnitude of the normalized correlation between the signals in the case of the standard inner products.

This concept can be extremely useful. For instance consider a situation in which we wish to determine which signal, if any, from a set X of signals most resembles a particular signal y. In order to accomplish this, we might evaluate the above expression for every signal $x \in X$, choosing the one that results in maxima provided that those maxima are above some threshold of "likeness". This is the idea behind the matched filter detector, which compares a set of signals against a target signal using the above expression in order to determine which is most like the target signal.

Matched Filter Detector Theory

Signal Comparison

The simplest variant of the matched filter detector scheme would be to find the member signal in a set X of signals that most closely matches a target signal y. Thus, for every $x \in X$ we wish to evaluate

$$m(x,y) = \left|\left\langle \frac{x}{||x||}, \frac{y}{\|y\|} \right\rangle\right|$$

in order to compare every member of X to the target signal y. Since the member of X which most closely matches the target signal y is desired, ultimately we wish to evaluate

$$x_m = \text{argmax}_{x \in X} \left|\left\langle \frac{x}{\|x\|}, \frac{y}{\|y\|} \right\rangle\right|$$

Note that the target signal does not technically need to be normalized to produce a maximum, but gives the desirable property that $m(x,y)$ is bounded to $[0,1]$.

The element $x_m \in X$ that produces the maximum value of $m(x,y)$ is not necessarily unique, so there may be more than one matching signal in X. Additionally, the signal $x_m \in X$ producing the maximum value of $m(x,y)$ may not produce a very large value of $m(x,y)$ and thus not be very much like the target signal y. Hence, another matched filter scheme might identify the argument producing the maximum but only above a certain threshold, returning no matching signals in X if the maximum is below the threshold. There also may be a signal $x \in X$ that produces a large value of $m(x,y)$ and thus has a high degree of "likeness" to y but does not produce the maximum value of $m(x,y)$. Thus, yet another matched filter scheme might identify all signals in X producing local maxima that are above a certain threshold.

Example 13.4.1

For example, consider the target signal given in Figure 13.4.1 and the set of two signals given in Figure 13.4.2. By inspection, it is clear that the signal g_2 is most like the target signal f. However, to make that conclusion mathematically, we use the matched filter detector with the L_2 inner product. If we were to actually make the necessary computations, we would first normalize each signal and then compute the necessary inner products in order to compare the signals in X with the target signal f. We

would notice that the absolute value of the inner product for g_2 with f when normalized is greater than the absolute value of the inner product of g_1 with f when normalized, mathematically stated as

$$g_2 = \text{argmax}_{x \in \{g_1, g_2\}} \left| \left\langle \frac{x}{\|x\|}, \frac{f}{\|f\|} \right\rangle \right|$$

Template Signal

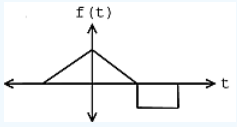
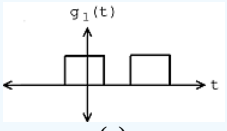
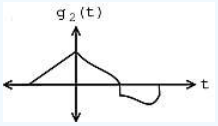


Figure 13.4.1: We wish to find a match for this target signal in the set of signals below.

Candidate Signals



(a)



(b)

Figure 13.4.2: We wish to find a match for the above target signal in this set of signals.

Pattern Detection

A somewhat more involved matched filter detector scheme would involve attempting to match a target time limited signal $y = f$ to a set of of time shifted and windowed versions of a single signal $X = \{wS_t g \mid t \in \mathbb{R}\}$ indexed by $\mathbb{R}$. The windowing funtion is given by $w(t) = u(t - t_1) - u(t - t_2)$ where $[t_1, t_2]$ is the interval to which f is time limited. This scheme could be used to find portions of g that have the same shape as f. If the absolute value of the inner product of the normalized versions of f and $wS_t g$ is large, which is the absolute value of the normalized correlation for standard inner products, then g has a high degree of "likeness" to f on the interval to which f is time limited but left shifted by t. Of course, if f is not time limited, it means that the entire signal has a high degree of "likeness" to f left shifted by t.

Thus, in order to determine the most likely locations of a signal with the same shape as the target signal f in a signal g we wish to compute

$$t_m = \text{argmax}_{t \in \mathbb{R}} \mid \left\langle \frac{f}{\|f\|}, \frac{wS_t g}{\|wS_t g\|} \right\rangle$$

to provide the desired shift. Assuming the inner product space examined is L_2 ($\mathbb{R}$ (similar results hold for $L_2(\mathbb{R}[a,b))$, $l_2(\mathbb{Z})$, and $l_2(\mathbb{Z}[a,b))$), this produces

$$t_m = \text{argmax}_{t \in \mathbb{R}} \left| \frac{1}{\|f\| \, \|wS_t g\|} \int_{-\infty}^{\infty} f(\tau) w(\tau) \overline{g(\tau - t)} d\tau \right|$$

Since f and w are time limited to the same interval

$$t_m = \text{argmax}_{t \in \mathbb{R}} \left| \frac{1}{\|f\| \, \|wS_t g\|} \int_{t_1}^{t_2} f(\tau) \overline{g(\tau - t)} d\tau \right|$$

Making the substitution $h(t) = \overline{g(-t)}$,

$$t_m = \text{argmax}_{t \in \mathbb{R}} \left| \frac{1}{\|f\| \, \|wS_t g\|} \int_{t_1}^{t_2} f(\tau) h(t - \tau) d\tau \right| .$$

Noting that this expression contains a convolution operation

$$t_m = \text{argmax}_{t\in\mathbb{R}} \left| \frac{(f * h)(t)}{\|f\| \, \|wS_t g\|} \right|$$

where h is the conjugate of the time reversed version of g defined by $h(t) = \overline{g(-t)}$.

In the special case in which the target signal f is not time limited, w has unit value on the entire real line. Thus, the norm can be evaluated as $\|wS_t g\| = \|S_t g\| = \|g\| = \|h\|$. Therefore, the function reduces to $t_m = \text{argmax}_{t\in\mathbb{R}} \frac{(f*h)(t)}{\|f\|\|h\|}$ where $h(t) = \overline{g(-t)}$. The function $f * g = \frac{(f*h)(t)}{\|f\|\|h\|}$ is known as the normalized cross-correlation of f and g.

Hence, this matched filter scheme can be implemented as a convolution. Therefore, it may be expedient to implement it in the frequency domain. Similar results hold for the $L_2(\mathbb{R}[a,b))$, $l_2(\mathbb{Z})$, and $l_2(\mathbb{Z}[a,b])$ spaces. It is especially useful to implement the $l_2(\mathbb{Z}[a,b])$ cases in the frequency domain as the power of the Fast Fourier Transform algorithm can be leveraged to quickly perform the computations in a computer program. In the $L_2(\mathbb{R}[a,b))$ and $l_2(\mathbb{Z}[a,b])$ cases, care must be taken to zero pad the signal if wrap-around effects are not desired. Similar results also hold for spaces on higher dimensional intervals with the same inner products.

Of course, there is not necessarily exactly one instance of a target signal in a given signal. There could be one instance, more than one instance, or no instance of a target signal. Therefore, it is often more practical to identify all shifts corresponding to local maxima that are above a certain threshold.

Example 13.4.2

The signal in Figure 13.4.4 contains an instance of the template signal seen in Figure 13.4.3 beginning at time $t = s_1$ as shown by the plot in Figure 13.4.5. Therefore,

$$s_1 = \text{argmax}_{t\in\mathbb{R}} \left| \left\langle \frac{f}{\|f\|}, \frac{wS_t g}{\|wS_t g\|} \right\rangle \right|.$$

Pattern Signal

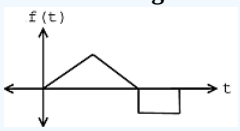


Figure 13.4.3: This function shows that pattern we are looking for in the signal below, which occurs at time $t = s_1$.

Longer Signal

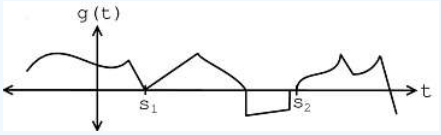


Figure 13.4.4: This signal contains an instance of the above signal starting at time $t = s_1$.

Absolute Value of Output

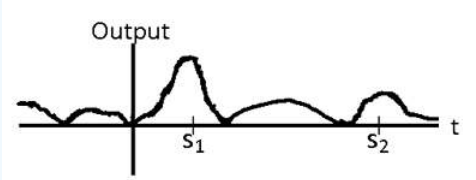


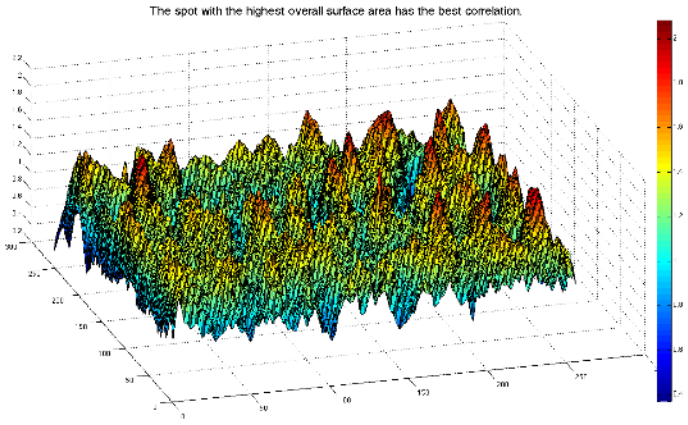
Figure 13.4.5: This signal shows a sketch of the absolute value of the matched filter output for the interval shown. Note that this was just an "eyeball approximation" sketch. Observe the pronounced peak at time $t = s_1$.

Practical Applications

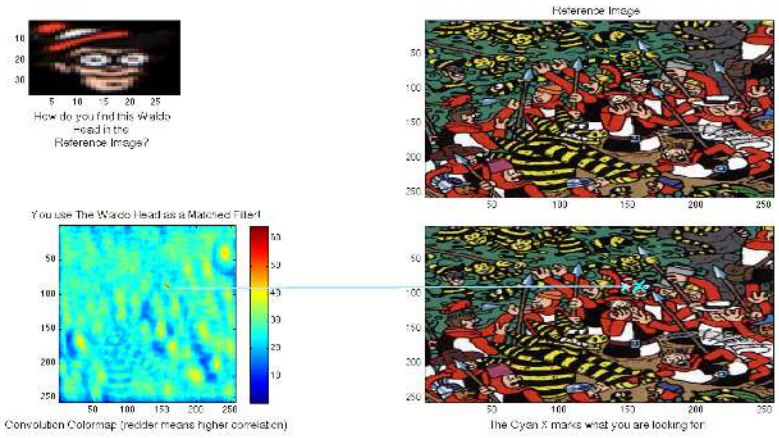
Image Detection

Matched Filtering is used in image processing to detect a template image within a reference image. This has real-word applications in verifying fingerprints for security or in verifying someone's photo. As a simple example, we can turn to the ever-popular "Where's Waldo?" books (known as Wally in the UK!), where the reader is tasked with finding the specific face of Waldo/Wally in a confusing background rife with look-alikes! If we are given the template head and a reference image, we can run a two dimensional convolution of the template image across the reference image to obtain a three dimensional convolution map, Figure

13.4.6(a), where the height of the convolution map is determined by the degree of correlation, higher being more correlated. Finding our target then becomes a matter of determining the spot where the local surface area is highest. The process is demonstrated in Figure 13.4.6(b). In the field of image processing, this matched filter-based process is known as **template matching**.



(a)



(b)

Figure 13.4.6: 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 Project at Rice University (click the link to learn more).

Exercise 13.4.1: Pros and Cons

What are the advantages of the matched filter algorithm to image detection? What are the drawbacks of this method?

Answer

This algorithm is very simple and thus easy to code. However, it is susceptible to certain types of noise - for example, it would be difficult to find Waldo if his face was rotated, flipped, larger or smaller than expected, or distorted in some other way.

Communications Systems

Matched filter detectors are also commonly used in Communications Systems. 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 Keying (FSK) where we having the following coding for '1' and '0':

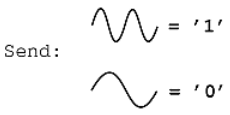


Figure 13.4.7: Frequency Shift Keying for '1' and '0'.

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 13.4.8:

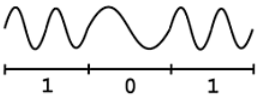


Figure 13.4.8: The bit stream "101" coded with the above FSK.

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 13.4.9:

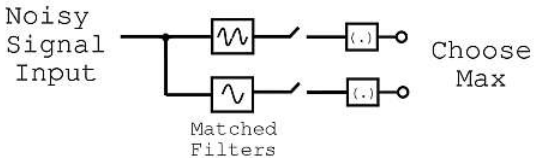


Figure 13.4.9: Block diagram of matched filter detector.

Figure 13.4.9 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.

The interactive example below supposes that our transmitter sends 1000 bits, plotting how many of those bits are received and interpreted correctly as either 1s and 0s, and also keeps a tally of how many are accidentally misinterpreted. You can play around with the distance between the energy of the "1" and the "0" (discriminability), the degree of noise present in the channel, and the location of the criterion (threshold) to get a feel for the basics of signal detection theory.

Example 13.4.3

Let's use a matched filter to find the "0" bits in a simple signal.

Let's use the signal $s_1(t)$ from Example 13.4.1 to represent the bits. $s_1(t)$ represents 0, while $-s_1(t)$ represents 1.

$0 \Rightarrow (b = 1) \Rightarrow (s_1(t) = s(t))$ for $0 \leq t \leq T$

$1 \Rightarrow (b = -1) \Rightarrow (s_2(t) = -s(t))$ for $0 \leq t \leq T$

$$X_t = \sum_{i=-P}^{P} b_i s(t - iT)$$

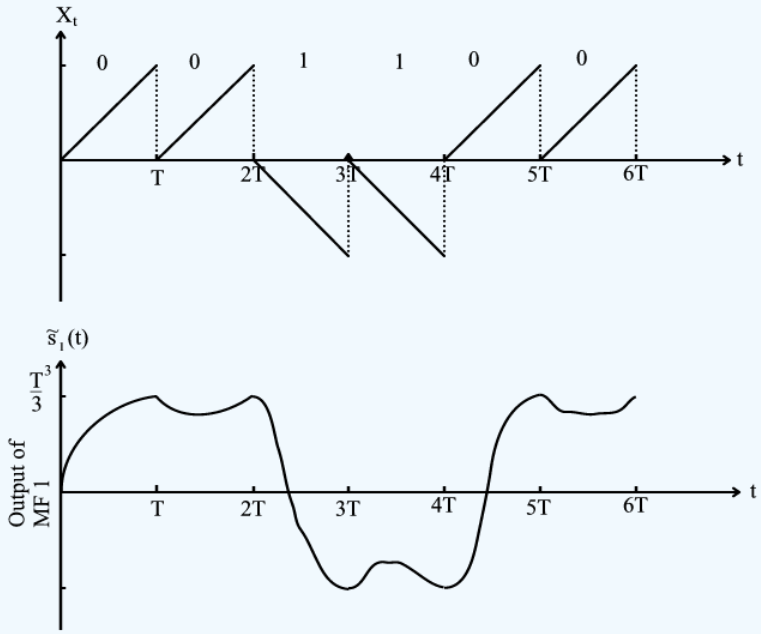


Figure 13.4.10

The matched filter output clearly shows the location of the "0" bits.

Radar

One of the first, and more intriguing forms of communication that used the matched filter concept was radar. A known electromagnetic signal is sent out by a transmitter at a target and reflected off of the target back to the sender with a time delay proportional to the distance between target and sender. This scaled, time-shifted signal is then convolved with the original template signal, and the time at which the output of this convolution is highest is noted.

This technology proved vital in the 1940s for the powers that possessed it. A short set of videos below shows the basics of how the technology works, its applications, and its impact in World War 2.

Figure 13.4.11

See the video in Figure 13.4.12 for an analysis of the same basic principle being applied to adaptive cruise control systems for the modern car.

Figure 13.4.12: Video on radar-based adaptive cruise control from The Science Channel.

Matched Filter Demonstration

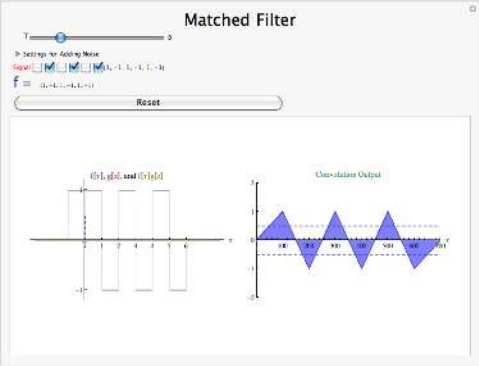


Figure 13.4.13: Interact (when online) with a Mathematica CDF demonstrating the Matched Filter. To Download, right-click and save target as .cdf.

Matched Filter Summary

As can be seen, the matched filter detector is an important signal processing application, rich both in theoretical concepts and in practical applications. The matched filter supports a wide array of uses related to pattern recognition, including image detection, frequency shift keying demodulation, and radar signal interpretation. Despite this diversity of purpose, all matched filter applications operate in essentially the same way. Every member of some set of signals is compared to a target signal by evaluating the absolute value of the inner product of the the two signals after normalization. However, the signal sets and result interpretations are application specific.

This page titled 13.4: Matched Filter Detector is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

14: Appendix A- Linear Algebra Overview

14.1: Basic Linear Algebra

14.2: Eigenvectors and Eigenvalues

14.3: Matrix Diagonalization

14.4: Eigen-Stuff in a Nutshell

14.5: Eigenfunctions of LTI Systems

This page titled 14: Appendix A- Linear Algebra Overview is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

14.1: Basic Linear Algebra

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.

Linear Independence

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

Definition: Linearly Independent

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

$$c_1 x_1 + c_2 x_2 + \cdots + c_n x_n = 0$$

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

Example 14.1.1

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 14.1.1), one can again prove that these vectors are **not** linearly independent.

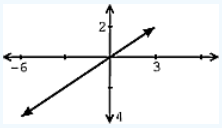


Figure 14.1.1: Graphical representation of two vectors that are not linearly independent.

Example 14.1.2

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 14.1.2) to check for linear independence.

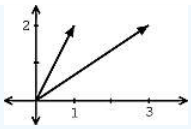


Figure 14.1.2: Graphical representation of two vectors that are linearly independent.

Exercise 14.1.1

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

$$x_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$
$$x_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$
$$x_3 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

Answer

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**!

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 14.1.3. Probably not, which is why the method used in the above solution becomes important.

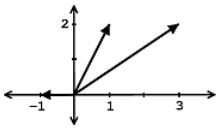


Figure 14.1.3: Plot of the three vectors. Can be shown that a linear combination exists among the three, and therefore they are **not** linear independent.

Note

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

Span

Definition: Word

The span 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 $\{x_1, x_2, \dots, x_k\}$

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

Example 14.1.3

Given the vector

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

the span of x_1 is a **line**.

Example 14.1.4

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 $\mathbb{C}^2$.

Basis

Definition: 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 14.1.5

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 ith place and the remaining values are zero. Then the **basis** for $\mathbb{C}^n$ is

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

Note

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

Example 14.1.6

$$h_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$h_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

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

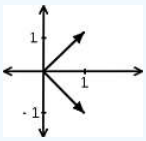


Figure 14.1.4: Plot of basis for $\mathbb{C}^2$

If $\{b_1, \dots, b_2\}$ is a basis for $\mathbb{C}^n$, then we can express **any** $x \in \mathbb{C}^n$ as a linear combination of the b_i's:

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

Example 14.1.7

Given the following vector,

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

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

$$x = e_1 + 2e_2$$

Exercise 14.1.2

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

Answer

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

In the two basis examples above, 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, which tells us that we can express periodic functions, $f(t)$, in terms of their basis functions, $e^{j\omega_0 nt}$.

Khan Lecture on Basis of a Subspace

Figure 14.1.5: video from Khan Academy, Basis of a Subspace - 20 min.

This page titled 14.1: Basic Linear Algebra is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

14.2: Eigenvectors and Eigenvalues

In this section, our linear systems will be n×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 14.1).

Eigenvectors and Eigenvalues

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

$$Ax = b \tag{14.2.1}$$

where x and b are $n \times 1$ vectors (Figure 14.2.1).

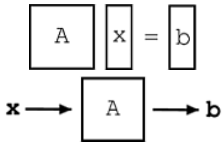


Figure 14.2.1: Illustration of linear system and vectors.

Definition: Eigenvector

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

$$A\mathbf{v} = \lambda\mathbf{v} \tag{14.2.2}$$

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

Graphical Model

Through Figure 14.2.2 and Figure 14.2.3, let us look at the difference between Equation 14.2.1 and Equation 14.2.2.

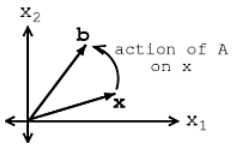


Figure 14.2.2: Represents Equation 14.2.1, $A\mathbf{x} = \mathbf{b}$.

If $\mathbf{v}$ is an eigenvector of A, then only its length changes. See Figure 14.2.3 and notice how our vector's length is simply scaled by our variable, λ, called the **eigenvalue**:

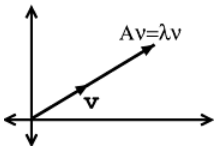


Figure 14.2.3: Represents Equation 14.2.2, $A\mathbf{v} = \lambda\mathbf{v}$.

Note

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

Examples

Exercise 14.2.1

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.

Answer

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

$$\lambda_1 = 3$$
$$\lambda_2 = -1$$

Exercise 14.2.2

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.

Answer

In order to prove that these two vectors are eigenvectors, we will show that these statements meet the requirements stated in the definition.

$$Av_1 = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$
$$Av_2 = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 4 \\ -4 \end{pmatrix}$$

These results show us that A only scales the two vectors (i.e. changes their length) and thus it proves that Equation 14.2.2 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.

Khan Lecture on Eigenvectors

Figure 14.2.4: video from Khan Academy - Introduction to Eigenvectors and Eigenvalues - 7:43 min.

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.

Finding Eigenvalues

Find $\lambda \in \mathbb{C}$ such that $\mathbf{v} \neq 0$, where 0 is the "zero vector." We will start with Equation 14.2.2, 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 14.2.1

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

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$
$$A - \lambda I = \begin{pmatrix} a_{1,1} - \lambda & a_{1,2} \\ a_{2,1} & a_{2,2} - \lambda \end{pmatrix}$$

If $(A - \lambda I)\mathrm{v} = 0$ for some $\mathrm{v} \neq 0$, 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 14.2.2

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 14.2.3

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

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$
$$A - \lambda I = \begin{pmatrix} a_{1,1} - \lambda & a_{1,2} \\ a_{2,1} & a_{2,2} - \lambda \end{pmatrix}$$
$$\det(A - \lambda I) = \lambda^2 - (a_{1,1} + a_{2,2})\,\lambda - a_{2,1}a_{1,2} + a_{1,1}a_{2,2}$$

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

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

Conclusion

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

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.

Khan Lecture on Deriving Eigenvectors and Eigenvalues

Figure 14.2.5: video from Khan Academy - Example Deriving Eigenvectors and Eigenvalues - 5:39 min.

Main Point

Say the eigenvectors of A, $\{v_1, v_2, \dots, v_n\}$, span $\mathbb{C}^n$, meaning $\{v_1, v_2, \dots, v_n\}$ are linearly independent (Section 14.1) and we can write any $\mathbf{x} \in \mathbb{C}^n$ as

$$\mathbf{x} = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n \tag{14.2.3}$$

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

$$\begin{aligned} A\mathbf{x} &= A\left(\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n\right) \\ A\mathbf{x} &= \alpha_1 A v_1 + \alpha_2 A v_2 + \cdots + \alpha_n A v_n \\ A\mathbf{x} &= \alpha_1 \lambda_1 v_1 + \alpha_2 \lambda_2 v_2 + \cdots + \alpha_n \lambda_n v_n = b \end{aligned}$$

Therefore we can write,

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

and this leads us to the following depicted system:



Figure 14.2.6: Depiction of system where we break our vector, x, into a sum of its eigenvectors.

where in Figure 14.2.6 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.

Practice Problem

Exercise 14.2.3

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

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

Answer

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 Equation 14.2.3, we want to represent x as a sum of its scaled eigenvectors. For this case, we have:

$$\mathbf{x} = 4v_1 + v_2$$
$$\mathbf{x} = \begin{pmatrix} 5 \\ 3 \end{pmatrix} = 4\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$
$$A\mathbf{x} = A\left(4v_1 + v_2\right) = \lambda_i\left(4v_1 + v_2\right)$$

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!

This page titled 14.2: Eigenvectors and Eigenvalues is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

14.3: Matrix Diagonalization

From our understanding of eigenvalues and eigenvectors (Section 14.2) 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 $\{v_1, v_2, \dots, v_n\}$?

Answer to Question #1

Question #1

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

If A has n **distinct** eigenvalues

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

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.

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

$$\mathbf{x} = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n \tag{14.3.1}$$

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 Equation 14.3.1 becomes

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

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.

Aside

Let us recall our knowledge of functions and their 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 x expressed in a different basis:

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

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

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

Matrix Diagonalization and Output

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

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

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

$$Ax = \begin{pmatrix} \vdots & \vdots & & \vdots \\ v_1 & v_2 & \ldots & v_n \\ \vdots & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} \lambda_1\alpha_1 \\ \vdots \\ \lambda_1\alpha_n \end{pmatrix}$$

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

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

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

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

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

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

Interpretation

For our interpretation, recall our key formulas:

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

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

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

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

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 14.3.1: Simple illustration of LTI system!

This page titled 14.3: Matrix Diagonalization is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

14.4: Eigen-Stuff in a Nutshell

A Matrix and its Eigenvector

The reason we are stressing eigenvectors (Section 14.2) and their importance is because the action of a matrix Aon one of its eigenvectors $\boldsymbol{v}$ is

1. extremely easy (and fast) to calculate

$$A\boldsymbol{v} = \lambda \boldsymbol{v} \tag{14.4.1}$$

   just **multiply** $\boldsymbol{v}$ by λ.

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

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

Using Eigenvectors' Span

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

$$\boldsymbol{x} = \alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_n v_n \tag{14.4.2}$$

Given Equation 14.4.2, we can rewrite $A\boldsymbol{x} = \boldsymbol{b}$. This equation is modeled in our LTI system pictured below:



Figure 14.4.1: LTI System.

$$\boldsymbol{x} = \sum_i \alpha_i v_i$$

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

The LTI system above represents our Equation 14.4.1. Below is an illustration of the steps taken to go from $\boldsymbol{x}$ to b.

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

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

1. Transform $\boldsymbol{x}$ 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 $\boldsymbol{b}$

This page titled 14.4: Eigen-Stuff in a Nutshell is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

14.5: Eigenfunctions of LTI Systems

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 14.4). We can develop the same ideas for LTI systems acting on signals. A linear time invariant (LTI) system $\mathcal{H}$ operating on a continuous input $f(t)$ to produce continuous time output $y(t)$

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

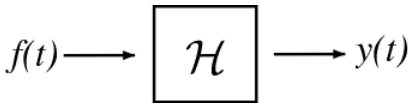


Figure 14.5.1: $\mathcal{H}[f(t)] = y(t)$. f and t are continuous time (CT) signals and $\mathcal{H}$ is an LTI operator.

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

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$



Figure 14.5.2: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ where $\boldsymbol{x}$ and $\boldsymbol{b}$ are in $\mathbb{C}^n$ and $\boldsymbol{A}$ is an $N \times N$ matrix.

Just as an eigenvector (Section 14.2) of $\boldsymbol{A}$ is a $\boldsymbol{v} \in \mathbb{C}^N$ such that $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{\lambda}\boldsymbol{v}$, $\boldsymbol{\lambda} \in \mathbb{C}$,



Figure 14.5.3: $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{\lambda}\boldsymbol{v}$ where $\boldsymbol{v} \in \mathbb{C}^N$ is an eigenvector of $\boldsymbol{A}$.

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

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

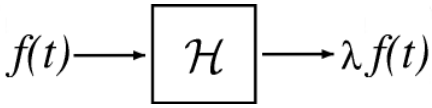


Figure 14.5.4: $\mathcal{H}[f(t)] = \lambda f(t), \quad \lambda \in \mathrm{C}$ where f is an eigenfunction of $\mathcal{H}$.

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.

Eigenfunctions of any LTI System

The class of LTI systems has a set of eigenfunctions in common: the complex exponentials (Section 1.8) e^{st}, $s \in \mathbb{C}$ are eigenfunctions for **all** LTI systems.

$$\mathcal{H}\left[e^{st}\right] = \lambda_s e^{st} \tag{14.5.1}$$

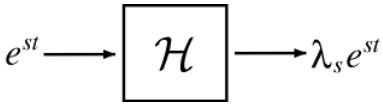


Figure 14.5.5: $\mathcal{H}\left[e^{st}\right] = \lambda_s e^{st}$ where $\mathcal{H}$ is an LTI system.

Note

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

We can prove Equation 14.5.1 by expressing the output as a convolution (Section 3.3) of the input e^{st} and the impulse response (Section 1.6) $h(t)$ of $\mathcal{H}$:

$$\begin{aligned}\mathcal{H}\left[e^{st}\right] &= \int_{-\infty}^{\infty} h(\tau)e^{s(t-\tau)}d\tau \\ &= \int_{-\infty}^{\infty} h(\tau)e^{st}e^{-(s\tau)}d\tau \\ &= e^{st}\int_{-\infty}^{\infty} h(\tau)e^{-(s\tau)}d\tau \end{aligned} \tag{14.5.2}$$

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

$$\mathcal{H}\left[e^{st}\right] = \lambda_s e^{st}$$

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 $H(s) \equiv \lambda_s$.

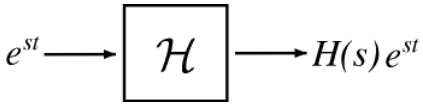


Figure 14.5.6: e^{st} is the eigenfunction and $H(s)$ are the eigenvalues.

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 gives us this representation for periodic continuous time signals, while the (slightly more complicated) Fourier transform lets us expand arbitrary continuous time signals.

This page titled 14.5: Eigenfunctions of LTI Systems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

15: Appendix B- Hilbert Spaces Overview

15.1: Fields and Complex Numbers
15.2: Vector Spaces
15.3: Norms
15.4: Inner Products
15.5: Hilbert Spaces
15.6: Cauchy-Schwarz Inequality
15.7: Common Hilbert Spaces
15.8: Types of Bases
15.9: Orthonormal Basis Expansions
15.10: Function Space
15.11: Haar Wavelet Basis
15.12: Orthonormal Bases in Real and Complex Spaces
15.13: Plancharel and Parseval's Theorems
15.14: Approximation and Projections in Hilbert Space

This page titled 15: Appendix B- Hilbert Spaces Overview is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.1: Fields and Complex Numbers

Fields

In order to propely discuss the concept of vector spaces in linear algebra, it is necessary to develop the notion of a set of "scalars" by which we allow a vector to be multiplied. A framework within which our concept of real numbers would fit is desireable. Thus, we would like a set with two associative, commutative operations (like standard addition and multiplication) and a notion of their inverse operations (like subtraction and division). The mathematical algebraic construct that addresses this idea is the field. A field $(S, +, *)$ is a set S together with two binary operations $+$ and $*$ such that the following properties are satisfied.

1. Closure of S under $+$: For every x, $y \in S$, $x + y \in S$.
2. Associativity of S under $+$: For every $x, y, z \in S$, $(x + y) + z = x + (y + z)$.
3. Existence of $+$ identity element: There is a $e_+ \in S$ such that for every $x \in S$, $e_+ + x = x + e_+ = x$.
4. Existence of $+$ inverse elements: For every $x \in S$ there is a $y \in S$ such that $x + y = y + x = e_+$.
5. Commutativity of S under $+$: For every $x, y \in S$, $x + y = y + x$.
6. Closure of S under $*$: For every $x, y \in S$, $x * y \in S$.
7. Associativity of S under $*$: For every $x, y, z \in S$, $(x * y) * z = x * (y * z)$.
8. Existence of $*$ identity element: There is a $e_* \in S$ such that for every $x \in S$, $e_* + x = x + e_* = x$.
9. Existence of $*$ inverse elements: For every $x \in S$ with $x \neq e_+$ there is a $y \in S$ such that $x * y = y * x = e_*$.
10. Commutativity of S under $*$: For every $x, y \in S$, $x * y = y * x$.
11. Distributivity of $*$ over $+$: For every $x, y, z \in S$, $x * (y + z) = xy + xz$.

While this definition is quite general, the two fields used most often in signal processing, at least within the scope of this course, are the real numbers and the complex numbers, each with their typical addition and multiplication operations.

The Complex Field

The reader is undoubtedly already sufficiently familiar with the real numbers with the typical addition and multiplication operations. However, the field of complex numbers with the typical addition and multiplication operations may be unfamiliar to some. For that reason and its importance to signal processing, it merits a brief explanation here.

Definitions

The notion of the square root of -1 originated with the quadratic formula: the solution of certain quadratic equations mathematically exists only if the so-called imaginary quantity $\sqrt{-1}$ could be defined. Euler first used i for the imaginary unit but that notation did not take hold until roughly Ampère's time. Ampère used the symbol i to denote current (intensité de current). It wasn't until the twentieth century that the importance of complex numbers to circuit theory became evident. By then, using i for current was entrenched and electrical engineers now choose j for writing complex numbers.

An **imaginary number** has the form $jb = \sqrt{-b^2}$. A **complex number**, z, consists of the ordered pair (a, b), a is the real component and b is the imaginary component (the j is suppressed because the imaginary component of the pair is always in the second position). The imaginary number jb equals $(0, b)$. Note that a and b are real-valued numbers.

Figure 15.1.1 shows that we can locate a complex number in what we call the **complex plane**. Here, a, the real part, is the x-coordinate and b, the imaginary part, is the y-coordinate.

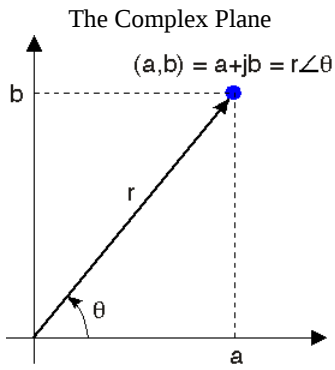


Figure 15.1.1: A complex number is an ordered pair (a, b) that can be regarded as coordinates in the plane. Complex numbers can also be expressed in polar coordinates as $r\angle\theta$.

From analytic geometry, we know that locations in the plane can be expressed as the sum of vectors, with the vectors corresponding to the x and y directions. Consequently, a complex number z can be expressed as the (vector) sum $z = a + jb$ where j indicates the y-coordinate. This representation is known as the **Cartesian form of z**. An imaginary number can't be numerically added to a real number; rather, this notation for a complex number represents vector addition, but it provides a convenient notation when we perform arithmetic manipulations.

The **real part** of the complex number $z = a + jb$, written as $\mathrm{Re}(z)$, equals a. We consider the real part as a function that works by selecting that component of a complex number not multiplied by j. The **imaginary part** of \(z\), $\mathrm{Im}(z)$, equals b: that part of a complex number that is multiplied by j. Again, both the real and imaginary parts of a complex number are real-valued.

The **complex conjugate** of z, written as z^*, has the same real part as z but an imaginary part of the opposite sign.

$$\begin{aligned} z &= \mathrm{Re}(z) + j\,\mathrm{Im}(z) \\ z^* &= \mathrm{Re}(z) - j\,\mathrm{Im}(z) \end{aligned} \tag{15.1.1}$$

Using Cartesian notation, the following properties easily follow.

- If we add two complex numbers, the real part of the result equals the sum of the real parts and the imaginary part equals the sum of the imaginary parts. This property follows from the laws of vector addition.

  $$a_1 + jb_1 + a_2 + jb_2 = a_1 + a_2 + j\,(b_1 + b_2)$$

  In this way, the real and imaginary parts remain separate.
- The product of j and a real number is an imaginary number: ja. The product of j and an imaginary number is a real number: $j(jb) = -b$ because $j^2 = -1$. Consequently, multiplying a complex number by j **rotates** the number's position by 90 degrees.

Exercise 15.1.1

Use the definition of addition to show that the real and imaginary parts can be expressed as a sum/difference of a complex number and its conjugate. $\mathrm{Re}(z) = \frac{z+z^*}{2}$ and $\mathrm{Im}(z) = \frac{z-z^*}{2j}$

Answer

$z + \bar{z} = a + jb + a - jb = 2a = 2\,\mathrm{Re}(z)$. Similarly, $z - \bar{z} = a + jb - (a - jb) = 2jb = 2(j, \mathrm{Im}(z))$

Complex numbers can also be expressed in an alternate form, **polar form**, which we will find quite useful. Polar form arises arises from the geometric interpretation of complex numbers. The Cartesian form of a complex number can be re-written as

$$a + jb = \sqrt{a^2 + b^2}\left(\frac{a}{\sqrt{a^2 + b^2}} + j\frac{b}{\sqrt{a^2 + b^2}}\right)$$

By forming a right triangle having sides a and b, we see that the real and imaginary parts correspond to the cosine and sine of the triangle's base angle. We thus obtain the **polar form** for complex numbers.

$$
\begin{aligned}
z &= a + jb = r\angle\theta \\
r &= |z| = \sqrt{a^2 + b^2} \\
a &= r\cos(\theta) \\
b &= r\sin(\theta) \\
\theta &= \arctan\left(\tfrac{b}{a}\right)
\end{aligned}
$$

The quantity r is known as the **magnitude** of the complex number z, and is frequently written as $|z|$. The quantity θ is the complex number's **angle**. In using the arc-tangent formula to find the angle, we must take into account the quadrant in which the complex number lies.

Exercise 15.1.2

Convert $3 - 2j$ to polar form.

Answer

To convert $3 - 2j$ to polar form, we first locate the number in the complex plane in the fourth quadrant. The distance from the origin to the complex number is the magnitude r, which equals $\sqrt{13} = \sqrt{3^2 + (-2)^2}$. The angle equals $-\arctan\left(\frac{2}{3}\right)$ or -0.588 radians (-33.7 degrees). The final answer is $\sqrt{13}\angle(-33.7)$ degrees.

Euler's Formula

Surprisingly, the polar form of a complex number z can be expressed mathematically as

$$z = r^{j\theta}$$

To show this result, we use **Euler's relations** that express exponentials with imaginary arguments in terms of trigonometric functions.

$$e^{j\theta} = \cos(\theta) + j\sin(\theta) \tag{15.1.2}$$

$$\cos(\theta) = \frac{e^{j\theta} + e^{-(j\theta)}}{2} \tag{15.1.3}$$

$$\sin(\theta) = \frac{e^{j\theta} - e^{-(j\theta)}}{2j}$$

The first of these is easily derived from the Taylor's series for the exponential.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

Substituting $j\theta$ for x, we find that

$$e^{j\theta} = 1 + j\frac{\theta}{1!} - \frac{\theta^2}{2!} - j\frac{\theta^3}{3!} + \ldots$$

because $j^2 = -1$, $j^3 = -j$, and $j^4 = 1$. Grouping separately the real-valued terms and the imaginary-valued ones,

$$e^{j\theta} = 1 - \frac{\theta^2}{2!} + \cdots + j\left(\frac{\theta}{1!} - \frac{\theta^3}{3!} + \ldots\right)$$

The real-valued terms correspond to the Taylor's series for $\cos(\theta)$, the imaginary ones to $\sin(\theta)$, and Euler's first relation results. The remaining relations are easily derived from the first. We see that multiplying the exponential in Equation 15.1.2 by a real constant corresponds to setting the radius of the complex number by the constant.

Calculating with Complex Numbers

Adding and subtracting complex numbers expressed in Cartesian form is quite easy: You add (subtract) the real parts and imaginary parts separately.

$$z_1 \pm z_2 = (a_1 \pm a_2) + j\,(b_1 \pm b_2)$$

To multiply two complex numbers in Cartesian form is not quite as easy, but follows directly from following the usual rules of arithmetic.

$$\begin{aligned} z_1 z_2 &= (a_1 + jb_1)(a_2 + jb_2) \\ &= a_1 a_2 - b_1 b_2 + j(a_1 b_2 + a_2 b_1) \end{aligned} \tag{15.1.4}$$

Note that we are, in a sense, multiplying two vectors to obtain another vector. Complex arithmetic provides a unique way of defining vector multiplication.

Exercise 15.1.3

What is the product of a complex number and its conjugate?

Answer

$z\bar{z} = (a + jb)(a - jb) = a^2 + b^2$. Thus $z\bar{z} = r^2 = (|z|)^2$.

Division requires mathematical manipulation. We convert the division problem into a multiplication problem by multiplying both the numerator and denominator by the conjugate of the denominator.

$$\begin{aligned} \frac{z_1}{z_2} &= \frac{a_1 + jb_1}{a_2 + jb_2} \\ &= \frac{a_1 + jb_1}{a_2 + jb_2}\frac{a_2 - jb_2}{a_2 - jb_2} \\ &= \frac{(a_1 + jb_1)(a_2 - jb_2)}{a_2^2 + b_2^2} \\ &= \frac{a_1 a_2 + b_1 b_2 + j(a_2 b_1 - a_1 b_2)}{a_2^2 + b_2^2} \end{aligned} \tag{15.1.5}$$

Because the final result is so complicated, it's best to remember **how** to perform division—multiplying numerator and denominator by the complex conjugate of the denominator—than trying to remember the final result.

The properties of the exponential make calculating the product and ratio of two complex numbers much simpler when the numbers are expressed in polar form.

$$\begin{aligned} z_1 z_2 &= r_1 e^{j\theta_1} r_2 e^{j\theta_2} \\ &= r_1 r_2 e^{j(\theta_1 + \theta_2)} \end{aligned} \tag{15.1.6}$$

$$\frac{z_1}{z_2} = \frac{r_1 e^{j\theta_2}}{r_2 e^{j\theta_2}} = \frac{r_1}{r_2} e^{j(\theta_1 - \theta_2)}$$

To multiply, the radius equals the product of the radii and the angle the sum of the angles. To divide, the radius equals the ratio of the radii and the angle the difference of the angles. When the original complex numbers are in Cartesian form, it's usually worth translating into polar form, then performing the multiplication or division (especially in the case of the latter). Addition and subtraction of polar forms amounts to converting to Cartesian form, performing the arithmetic operation, and converting back to polar form.

This page titled 15.1: Fields and Complex Numbers is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.2: Vector Spaces

Introduction

Definition: Vector Space

A 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 $alpha \in \mathbb{R}$, $\alpha \in \mathbb{C}$, or some other field) and (2) if $f_1 \in S$, $f_2 \in S$, then $(f_1 + f_2) \in S$

To define an vector space, we need

- A set of things called "vectors" (X)
- A set of things called "scalars" that form a field (A)
- A vector addition operation ()
- A scalar multiplication operation ($*$)

The operations need to have all the properties of given below. Closure is usually the most important to show.

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**

Properties

We define a set V to be a vector space if

1. $x + y = y + x$ for each x and y in V
2. $x + (y + z) = (x + y) + z$ for each x, y, and z in V
3. There is a unique "zero vector" such that $x + 0 = x$ for each x in V (0 is the field additive identity)
4. For each x in V there is a unique vector $-x$ such that $x + -x = 0$
5. $1x = x$ (1 is the field multiplicative identity)
6. $(c_1 c_2)x = c_1(c_2 x)$ for each x in V and c_1 and c_2 in $\mathbb{C}$
7. $c(x + y) = cx + cy$ for each x and y in V and c in $\mathbb{C}$
8. $(c_1 + c_2)x = c_1 x + c_2 x$ for each x in V and c_1 and c_2 in $\mathbb{C}$

Examples

- $\mathbb{R}^n$ = real vector space
- $\mathbb{C}^n$ = complex vector space
- $L^1(\mathbb{R}) = \left\{ f(t), f(t) \middle| \int_{-\infty}^{\infty} |f(t)| \mid dt < \infty \right\}$ is a vector space
- $L^\infty(\mathbb{R}) = \{ f(t), f(t) \mid f(t) \text{ is bounded } \}$ is a vector space
- $L^2(\mathbb{R}) = \left\{ f(t), f(t) \middle| \int_{-\infty}^{\infty} (|f(t)|)^2 dt < \infty \right\}$ = finite energy signals is a vector space
- $L^2([0, T])$ = 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

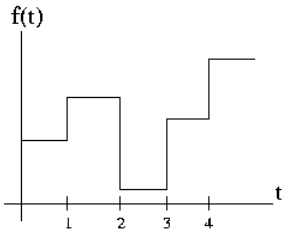


Figure 15.2.1

- $\mathbb{R}_+^2 = \left\{ \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \mid (x_0 > 0) \wedge (x_1 > 0) \right\}$ 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)

This page titled 15.2: Vector Spaces is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.3: Norms

Introduction

This module will explain norms, a mathematical concept that provides a notion of the size of a vector. Specifically, the general definition of a norm will be discussed and discrete time signal norms will be presented.

Norms

The **norm** of a vector is a real number that represents the "size" of the vector.

Example 15.3.1

In $\mathbb{R}^2$, we can define a norm to be a vectors geometric length.

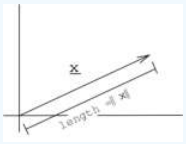


Figure 15.3.1

$\boldsymbol{x} = (x_0, x_1)^T$, norm $\|\boldsymbol{x}\| = \sqrt{x_0^2 + x_1^2}$

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 $\|x\| > 0$, $x \in S$
2. scaling a vector scales the norm by the same amount $\|\alpha x\| = |\alpha|\|x\|$ for all vectors x and scalars α
3. Triangle Property: $\|x + y\| \leq \|x\| + \|y\|$ for all vectors x, y. "The "size" of the sum of two vectors is less than or equal to the sum of their sizes"

A vector space (Section 15.2) with a well defined norm is called a **normed vector space** or **normed linear space**.

Examples

Example 15.3.2

$\mathbb{R}^n$ (or $\mathbb{C}^n$), $\boldsymbol{x} = \begin{pmatrix} x_0 \\ x_1 \\ \cdots \\ x_{n-1} \end{pmatrix}$, $\|x\|_1 = \sum_{i=0}^{n-1} |x_i|$, $\mathbb{R}^n$ with this norm is called $\ell^1([0, n-1])$.

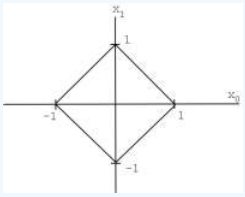


Figure 15.3.2: Collection of all $\boldsymbol{x} \in \mathbb{R}^2$ with $\|x\|_1 = 1$

Example 15.3.3

$\mathbb{R}^n$ (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).

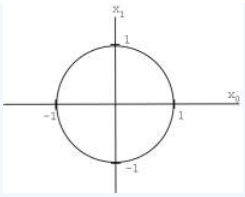


Figure 15.3.3: Collection of all $\boldsymbol{x} \in \mathbb{R}^2$ with $\|x\|_2 = 1$

Example 15.3.4

$\mathbb{R}^n$ (or $\mathbb{C}^n$), with norm $\|x\|_\infty = \max_i \{i, |x_i|\}$ is called $\ell^\infty([0, n-1])$

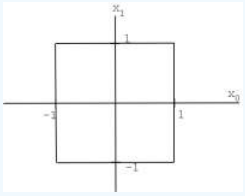


Figure 15.3.4: $x \in \mathbb{R}^2$ with $\|x\|_\infty = 1$

Spaces of Sequences and Functions

We can define similar norms for spaces of sequences and functions.

Discrete time signals = sequences of numbers

$$x[n] = \{\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots\}$$

- $\|x(n)\|_1 = \sum_{i=-\infty}^{\infty} |x[i]|, x[n] \in \ell^1(\mathbb{Z}) \Rightarrow (\|x\|_1 < \infty)$
- $\|x(n)\|_2 = (\sum_{i=-\infty}^{\infty} (|x[i]|)^2)^{\frac{1}{2}}, x[n] \in \ell^2(\mathbb{Z}) \Rightarrow (\|x\|_2 < \infty)$
- $\|x(n)\|_p = (\sum_{i=-\infty}^{\infty} (|x[i]|)^P)^{\frac{1}{p}}, x[n] \in \ell^p(\mathbb{Z}) \Rightarrow (\|x\|_p < \infty)$
- $\|x(n)\|_\infty = \sup_i |x[i]|, x[n] \in \ell^\infty(\mathbb{Z}) \Rightarrow (\|x\|_\infty < \infty)$

For continuous time functions:

- $\|f(t)\|_p = \left(\int_{-\infty}^{\infty} (|f(t)|)^p dt\right)^{\frac{1}{p}}, f(t) \in L^p(\mathbb{R}) \Rightarrow (\|f(t)\|_p < \infty)$
- $\|f(t)\|_p = \left(\int_0^T (|f(t)|)^p dt\right)^{\frac{1}{p}}, f(t) \in L^p([0, T]) \Rightarrow (\|f(t)\|_p < \infty)$

This page titled 15.3: Norms is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.4: Inner Products

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 $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$.

Definition: Standard Inner Product

The standard inner product is defined mathematically as:

$$\begin{aligned}\langle \boldsymbol{x}, \boldsymbol{y} \rangle &= \boldsymbol{y}^T \boldsymbol{x} \\ &= \begin{pmatrix} y_0 & y_1 & \cdots & y_{n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} \\ &= \sum_{i=0}^{n-1} x_i y_i\end{aligned}$$

Inner Product in 2-D

If we have$\boldsymbol{x} \in \mathbb{R}^2$ and $\boldsymbol{y} \in \mathbb{R}^2$, then we can write the inner product as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \|\boldsymbol{x}\| \|\boldsymbol{y}\| \cos(\theta)$$

where θ is the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$.

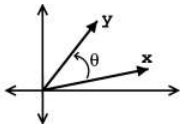


Figure 15.4.1: General plot of vectors and angle referred to in above equations.

Geometrically, the inner product tells us about the **strength** of $\boldsymbol{x}$ in the direction of $\boldsymbol{y}$.

Example 15.4.1

For example, if $\|x\| = 1$, then

$$< x, y >= \|y\| \cos(\theta)$$

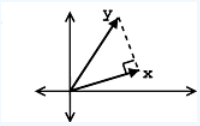


Figure 15.4.2: Plot of two vectors from above example.

The following characteristics are revealed by the inner product:

- $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ measures the length of the **projection** of $\boldsymbol{y}$ onto $\boldsymbol{x}$.
- $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ is **maximum** (for given $\|\boldsymbol{x}\|$, $\|\boldsymbol{y}\|$) when $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the same direction ($(\theta = 0) \Rightarrow (\cos(\theta) = 1)$).
- $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ is zero when $(\cos(\theta) = 0) \Rightarrow (\theta = 90^\circ)$, i.e. $\boldsymbol{x}$ and $\boldsymbol{y}$ are **orthogonal**.

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:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \overline{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}$$

- Scaling:

- Scaling:

$$\langle \alpha \boldsymbol{x}, \boldsymbol{y} \rangle = \alpha \langle (\boldsymbol{x}, \boldsymbol{y}) \rangle$$

- Additivity:

$$\langle \boldsymbol{x} + \boldsymbol{y}, \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{z} \rangle + \langle \boldsymbol{y}, \boldsymbol{z} \rangle$$

- "Positivity":

$$\langle \boldsymbol{x}, \boldsymbol{x} \rangle > 0, \boldsymbol{x} \neq 0$$

Definition: Orthogonal

We say that $\boldsymbol{x}$ and $\boldsymbol{y}$ are orthogonal if:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$$

This page titled 15.4: Inner Products is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.5: Hilbert Spaces

Hilbert Spaces

A vector space S with a valid inner product (Section 15.4) 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 Hilbert, who developed this idea through his studies of integral equations. We define our valid norm using the inner product as:

$$\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$$

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.

Examples of Hilbert Spaces

Below we will list a few examples of Hilbert spaces. You can verify that these are valid inner products at home.

- For $\mathbb{C}^n$,

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{y}^T \boldsymbol{x} = \begin{pmatrix} \overline{y_0} & \overline{y_1} & \dots & \overline{y_{n-1}} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} = \sum_{i=0}^{n-1} x_i \overline{y_i}$$

- Space of finite energy complex functions: $L^2(\mathbb{R})$

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle = \int_{-\infty}^{\infty} f(t) \overline{g(t)} \mathrm{d}t$$

- Space of square-summable sequences: $\ell^2(\mathbb{Z})$

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=-\infty}^{\infty} x[i] \overline{y[i]}$$

This page titled 15.5: Hilbert Spaces is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.6: Cauchy-Schwarz Inequality

Introduction

Any treatment of linear algebra as relates to signal processing would not be complete without a discussion of the Cauchy-Schwarz inequality, a relation that enables a wide array of signal processing applications related to pattern matching through a method called the matched filter. Recall that in standard Euclidean space, the angle θ between two vectors x, y is given by

$$\cos(\theta) = \frac{\langle x, y \rangle}{\|x\|\|y\|}.$$

Since $\cos(\theta) \leq 1$, it follows that

$$|\langle x, y \rangle|^2 \leq \langle x, x \rangle \langle y, y \rangle.$$

Furthermore, equality holds if and only if $\cos(\theta) = 0$, implying that

$$|\langle x, y \rangle|^2 = \langle x, x \rangle \langle y, y \rangle$$

if and only if $y = ax$ for some real a. This relation can be extended to all inner product spaces over a real or complex field and is known as the Cauchy-Schwarz inequality, which is of great importance to the study of signals.

The Cauchy-Schwarz Inequality

Statement of the Cauchy-Schwarz Inequality

The general statement of the Cauchy-Schwarz inequality mirrors the intuition for standard Euclidean space. Let V be an inner product space over the field of complex numbers $\mathbb{C}$ with inner product $\langle \cdot, \cdot \rangle$. For every pair of vectors $x, y \in V$ the inequality

$$|\langle x, y \rangle|^2 \leq \langle x, x \rangle \langle y, y \rangle$$

holds. Furthermore, the equality

$$|\langle x, y \rangle|^2 = \langle x, x \rangle \langle y, y \rangle$$

holds if and only if $y = ax$ for some $a \in \mathbb{C}$. That is, equality holds if and only if x and y are linearly dependent.

Proof of the Cauchy-Schwarz Inequality

Let V be a vector space over the real or complex field F, and let $x, y \in V$ be given. In order to prove the Cauchy-Schwarz inequality, it will first be proven that $|\langle x, y \rangle|^2 = \langle x, x \rangle \langle y, y \rangle$ if $y = ax$ for some $a \in F$. It will then be shown that $|\langle x, y \rangle|^2 < \langle x, x \rangle \langle y, y \rangle$ if $y \neq ax$ for all $a \in F$.

Consider the case in which $y = ax$ for some $a \in F$. From the properties of inner products, it is clear that

$$\begin{aligned} |\langle x, y \rangle|^2 &= |\langle x, ax \rangle|^2 \\ &= |\bar{a}\langle x, x \rangle|^2 \end{aligned} \tag{15.6.1}$$

Hence, it follows that

$$\begin{aligned} |\langle x, y \rangle|^2 &= |\bar{a}|^2 |\langle x, x \rangle|^2 \\ &= |a|^2 \langle x, x \rangle^2 \end{aligned} \tag{15.6.2}$$

Similarly, it is clear that

$$\begin{aligned} \langle x, x \rangle \langle y, y \rangle &= \langle x, x \rangle \langle ax, ax \rangle \\ &= \langle x, x \rangle a\bar{a} \langle x, x \rangle \\ &= |a|^2 \langle x, x \rangle^2 \end{aligned} \tag{15.6.3}$$

Thus, it is proven that $|\langle x, y \rangle|^2 = \langle x, x \rangle \langle y, y \rangle$ if $x = ay$ for some $a \in F$.

Next, consider the case in which $y \neq ax$ for all $a \in F$, which implies that $y \neq 0$ so $\langle y, y \rangle \neq 0$. Thus, it follows by the properties of inner products that, for all $a \in F$, $\langle x - ay, x - ay \rangle > 0$. This can be expanded using the properties of inner products to the expression

$$\begin{aligned}\langle x-ay, x-ay\rangle &= \langle x, x-ay\rangle - a\langle y, x-ay\rangle \\ &= \langle x,x\rangle - \bar{a}\langle x,y\rangle - a\langle y,x\rangle + |a|^2\langle y,y\rangle \end{aligned} \tag{15.6.4}$$

Choosing $a = \frac{\langle x,y\rangle}{\langle y,y\rangle}$,

$$\begin{aligned}\langle x-ay, x-ay\rangle &= \langle x,x\rangle - \frac{\langle y,x\rangle}{\langle y,y\rangle}\langle x,y\rangle - \frac{\langle x,y\rangle}{\langle y,y\rangle}\langle y,x\rangle + \frac{\langle x,y\rangle\langle y,x\rangle}{\langle y,y\rangle^2}\langle y,y\rangle \\ &= \langle x,x\rangle - \frac{\langle x,y\rangle\langle y,x\rangle}{\langle y,y\rangle}\end{aligned} \tag{15.6.5}$$

Hence, it follows that $\langle x,x\rangle - \frac{\langle x,y\rangle\langle y,x\rangle}{\langle y,y\rangle} > 0$. Consequently, $\langle x,x\rangle\langle y,y\rangle - \langle x,y\rangle\overline{\langle x,y\rangle} > 0$. Thus, it can be concluded that $|\langle x,y\rangle|^2 < \langle x,x\rangle\langle y,y\rangle$ if $y \neq ax$ for all $a \in F$.

Therefore, the inequality

$$|\langle x,y\rangle|^2 \leq \langle x,x\rangle\langle y,y\rangle$$

holds for all $x, y \in V$, and equality

$$|\langle x,y\rangle|^2 = \langle x,x\rangle\langle y,y\rangle$$

holds if and only if $y = ax$ for some $a \in F$.

Additional Mathematical Implications

Consider the maximization of $\left|\left\langle \frac{x}{\|x\|}, \frac{y}{\|y\|}\right\rangle\right|$ where the norm $\|\cdot\| = \langle\cdot,\cdot\rangle$ is induced by the inner product. By the Cauchy-Schwarz inequality, we know that $\left|\left\langle \frac{x}{||x||}, \frac{y}{\|y\|}\right\rangle\right|^2 \leq 1$ and that $\left|\left\langle \frac{x}{||x||}, \frac{y}{\|y\|}\right\rangle\right|^2 = 1$ if and only if $\frac{y}{\|y\|} = a\frac{x}{\|x\|}$ for some $a \in \mathbb{C}$. Hence, $\left|\left\langle \frac{x}{||x||}, \frac{y}{\|y\|}\right\rangle\right|$ attains a maximum where $\frac{y}{\|y\|} = a\frac{x}{\|x\|}$ for some $a \in \mathbb{C}$. Thus, collecting the scalar variables, $\left|\left\langle \frac{x}{||x||}, \frac{y}{\|y\|}\right\rangle\right|$ attains a maximum where $y = ax$. This result will be particularly useful in developing the matched filter detector techniques.

Matched Filter Detector

Background Concepts

A great many applications in signal processing, image processing, and beyond involve determining the presence and location of a target signal within some other signal. A radar system, for example, searches for copies of a transmitted radar pulse in order to determine the presence of and distance to reflective objects such as building or aircraft. A communication system searches for copies of waveforms representing digital 0s and 1s in order to receive a message.

As has already been shown, the expression $\left|\left\langle \frac{x}{\|x\|}, \frac{y}{\|y\|}\right\rangle\right|$ attains its upper bound, which is 1, when $y = ax$ for some scalar a in a real or complex field. The lower bound, which is 0, is attained when x and y are orthogonal. In informal intuition, this means that the expression is maximized when the vectors x and y have the same shape or pattern and minimized when x and y are very different. A pair of vectors with similar but unequal shapes or patterns will produce relatively large value of the expression less than 1, and a pair of vectors with very different but not orthogonal shapes or patterns will produce relatively small values of the expression greater than 0. Thus, the above expression carries with it a notion of the degree to which two signals are “alike”, the magnitude of the normalized correlation between the signals in the case of the standard inner products.

This concept can be extremely useful. For instance consider a situation in which we wish to determine which signal, if any, from a set X of signals most resembles a particular signal y. In order to accomplish this, we might evaluate the above expression for every signal $x \in X$, choosing the one that results in maxima provided that those maxima are above some threshold of “likeness”. This is the idea behind the matched filter detector, which compares a set of signals against a target signal using the above expression in order to determine which among them are most like the target signal. For a detailed treatment of the applications of the matched filter detector see the liked module.

Signal Comparison

The simplest variant of the matched filter detector scheme would be to find the member signal in a set X of signals that most closely matches a target signal y. Thus, for every $x \in X$ we wish to evaluate

$$m(x,y) = \left| \left\langle \frac{x}{\|x\|}, \frac{y}{\|y\|} \right\rangle \right|$$

in order to compare every member of X to the target signal y. Since the member of X which most closely matches the target signal y is desired, ultimately we wish to evaluate

$$x_m = \text{argmax}_{x \in X} \left| \left\langle \frac{x}{\|x\|}, \frac{y}{\|y\|} \right\rangle \right|.$$

Note that the target signal does not technically need to be normalized to produce a maximum, but gives the desirable property that $m(x,y)$ is bounded to $[0,1]$.

The element $x_m \in X$ that produces the maximum value of $m(x,y)$ is not necessarily unique, so there may be more than one matching signal in X. Additionally, the signal $x_m \in X$ producing the maximum value of $m(x,y)$ may not produce a very large value of $m(x,y)$ and thus not be very much like the target signal y. Hence, another matched filter scheme might identify the argument producing the maximum but only above a certain threshold, returning no matching signals in X if the maximum is below the threshold. There also may be a signal $x \in X$ that produces a large value of $m(x,y)$ and thus has a high degree of "likeness" to yy but does not produce the maximum value of $m(x,y)$. Thus, yet another matched filter scheme might identify all signals in X producing local maxima that are above a certain threshold.

Example 15.6.1

For example, consider the target signal given in Figure 15.6.1 and the set of two signals given in Figure 15.6.2. By inspection, it is clear that the signal g_2 is most like the target signal f. However, to make that conclusion mathematically, we use the matched filter detector with the L_2 inner product. If we were to actually make the necessary computations, we would first normalize each signal and then compute the necessary inner products in order to compare the signals in X with the target signal f. We would notice that the absolute value of the inner product for g_2 with f when normalized is greater than the absolute value of the inner product of g_1 with f when normalized, mathematically stated as

$$g_2 = \text{argmax}_{x \in \{g_1, g_2\}} \left| \left\langle \frac{x}{\|x\|}, \frac{f}{\|f\|} \right\rangle \right|$$

Template Signal

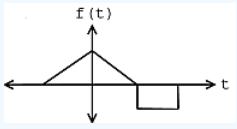


Figure 15.6.1: We wish to find a match for this target signal in the set of signals below.

Candidate Signals

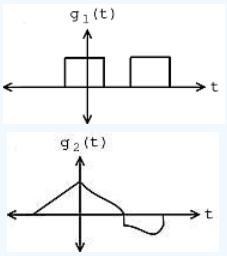


Figure 15.6.2: We wish to find a match for the above target signal in this set of signals.

Pattern Detection

A somewhat more involved matched filter detector scheme would involve attempting to match a target time limited signal $y = f$ to a set of of time shifted and windowed versions of a single signal $X = \{wS_t g \mid t \in \mathbb{R}\}$ indexed by $\mathbb{R}$. The windowing function is given by $w(t) = u(t-t_1) - u(t-t_2)$ where $[t_1, t_2]$ is the interval to which f is time limited. This scheme could be used to find portions of g that have the same shape as f. If the absolute value of the inner product of the normalized versions of f and $wS_t g$ is large, which is the absolute value of the normalized correlation for standard inner products, then gg has a high degree of "likeness" to f on the

interval to which f is time limited but left shifted by t. Of course, if f is not time limited, it means that the entire signal has a high degree of "likeness" to f left shifted by t.

Thus, in order to determine the most likely locations of a signal with the same shape as the target signal f in a signal g we wish to compute

$$t_m = \text{argmax}_{t\in\mathbb{R}} \left| \left\langle \frac{f}{\|f\|}, \frac{wS_t g}{\|wS_t g\|} \right\rangle \right|$$

to provide the desired shift. Assuming the inner product space examined is $L_2(\mathbb{R}$ (similar results hold for $L_2(\mathbb{R}[a,b))$, $l_2(\mathbb{Z})$, and $l_2(\mathbb{Z}[a,b)))$, this produces

$$t_m = \text{argmax}_{t\in\mathbb{R}} \mid \frac{1}{\|f\| \, \|wS_t g\|} \int_{-\infty}^{\infty} f(\tau) w(\tau) \overline{g(\tau - t)} d\tau$$

Since f and w are time limited to the same interval

$$t_m = \text{argmax}_{t\in\mathbb{R}} \left| \frac{1}{\|f\| \, \|wS_t g\|} \int_{t_1}^{t_2} f(\tau) \overline{g(\tau - t)} d\tau \right|$$

Making the substitution $h(t) = \overline{g(-t)}$,

$$t_m = \text{argmax}_{t\in\mathbb{R}} \left| \frac{1}{\|f\| \, \|wS_t g\|} \int_{t_1}^{t_2} f(\tau) h(t - \tau) d\tau \right|$$

Noting that this expression contains a convolution operation

$$t_m = \text{argmax}_{t\in\mathbb{R}} \left| \frac{(f * h)(t)}{\|f\| \, \|wS_t g\|} \right| .$$

where h is the conjugate of the time reversed version of g defined by $h(t) = \overline{g(-t)}$.

In the special case in which the target signal f is not time limited, w has unit value on the entire real line. Thus, the norm can be evaluated as $\|wS_t g\| = \|S_t g\| = \|g\| = \|h\|$. Therefore, the function reduces to $t_m = \text{argmax}_{t\in\mathbb{R}} \frac{(f^* h)(t)}{\|f\| \|h\|}$ where $h(t) = \overline{g(-t)}$. The function $f \star g = \frac{(f^* h)(t)}{\|f\| \|h\|}$ is known as the normalized cross-correlation of f and g.

Hence, this matched filter scheme can be implemented as a convolution. Therefore, it may be expedient to implement it in the frequency domain. Similar results hold for the $L_2(\mathbb{R}[a,b))$, $l_2(\mathbb{Z})$, and $l_2(\mathbb{Z}[a,b])$ spaces. It is especially useful to implement the $l_2(\mathbb{Z}[a,b])$ cases in the frequency domain as the power of the Fast Fourier Transform algorithm can be leveraged to quickly perform the computations in a computer program. In the $L_2(\mathbb{R}[a,b))$ and $l_2(\mathbb{Z}[a,b])$ cases, care must be taken to zero pad the signal if wrap-around effects are not desired. Similar results also hold for spaces on higher dimensional intervals with the same inner products.

Of course, there is not necessarily exactly one instance of a target signal in a given signal. There could be one instance, more than one instance, or no instance of a target signal. Therefore, it is often more practical to identify all shifts corresponding to local maxima that are above a certain threshold.

Example 15.6.2

The signal in Figure 15.6.4 contains an instance of the template signal seen in Figure 15.6.3 beginning at time $t = s_1$ as shown by the plot in Figure 15.6.5. Therefore,

$$s_1 = \text{argmax}_{t\in\mathbb{R}} \left| \left\langle \frac{f}{\|f\|}, \frac{wS_t g}{\|wS_t g\|} \right\rangle \right|$$

Pattern Signal

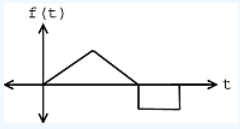


Figure 15.6.3: This function shows tha pattern we are looking for in the signal below, which occurs at time $t = s_1$.

Longer Signal

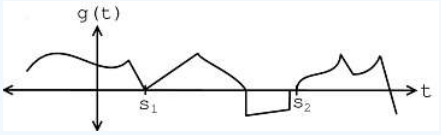


Figure 15.6.4: This signal contains an instance of the above signal starting at time $t = s_1$.

Absolute Value of Output

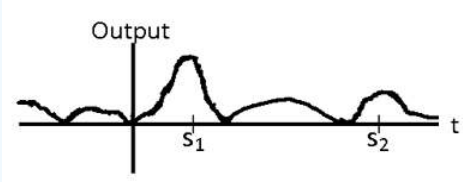


Figure 15.6.5: This signal shows a sketch of the absolute value of the matched filter output for the interval shown. Note that this was just an "eyeball approximation" sketch. Observe the pronounced peak at time $t = s_1$.

Cauchy-Schwarz Inequality Video Lecture

Proof of the Cauchy-Schwarz Inequality

Figure 15.6.6: Video lecture on the proof of the Cauchy-Schwarz inequality from Khan Academy. Only part of the theorem is proven.

Cauchy-Schwarz Inequality Summary

As can be seen, the Cauchy-Schwarz inequality is a property of inner product spaces over real or complex fields that is of particular importance to the study of signals. Specifically, the implication that the absolute value of an inner product is maximized over normal vectors when the two arguments are linearly dependent is key to the justification of the matched filter detector. Thus, it enables the use of matched filters for such pattern matching applications as image detection, communications demodulation, and radar signal analysis.

This page titled 15.6: Cauchy-Schwarz Inequality is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.7: Common Hilbert Spaces

Common Hilbert Spaces

Below we will look at the four most common Hilbert spaces (Section 15.4) that you will have to deal with when discussing and manipulating signals and systems.

$\mathbb{R}^n$ (real scalars) and $\mathbb{C}^n$ (complex scalars), also called $\ell^2([0, n-1])$

$\boldsymbol{x} = \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{n-1} \end{pmatrix}$ is a list of numbers (finite sequence). The inner product (Section 15.4) for our two spaces are as follows:

- Standard inner product $\mathbb{R}^n$:

$$\begin{aligned} \langle \boldsymbol{x}, \boldsymbol{y} \rangle &= \boldsymbol{y}^T \boldsymbol{x} \\ &= \sum_{i=0}^{n-1} x_i y_i \end{aligned} \tag{15.7.1}$$

- Standard inner product $\mathbb{C}^n$:

$$\begin{aligned} \langle \boldsymbol{x}, \boldsymbol{y} \rangle &= \overline{\boldsymbol{y}^T} \boldsymbol{x} \\ &= \sum_{i=0}^{n-1} x_i \bar{y}_i \end{aligned} \tag{15.7.2}$$

Model for: Discrete time signals on the interval $[0, n-1]$ **or** periodic (with period n) discrete time signals.

$$\begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{n-1} \end{pmatrix}$$

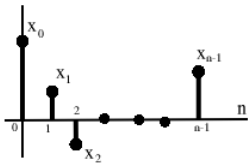


Figure 15.7.1

$f \in L^2([a, b])$ is a **finite energy** function on $[a, b]$

Inner Product

$$\langle f, g \rangle = \int_a^b f(t) \overline{g(t)} \mathrm{d}t$$

Model for: continuous time signals on the interval $[a, b]$ **or** periodic (with period $T = b - a$) continuous time signals

$x \in \ell^2(\mathbb{Z})$ is an **infinite sequence of numbers** that's square-summable

Inner product

$$\langle x, y \rangle = \sum_{i=-\infty}^{\infty} x[i] \overline{y[i]}$$

Model for: discrete time, non-periodic signals

$f \in L^2(\mathbb{R})$ is a **finite energy function** on all of $\mathbb{R}$.

Inner product

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(t)\overline{g(t)}\mathrm{d}t$$

Model for: continuous time, non-periodic signals

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

$$\begin{aligned} \langle f, f \rangle &= \left(\int (|f(t)|)^2 \mathrm{d}t \right)^2 \\ &= (\|f\|_1)^2 \end{aligned} \tag{15.7.3}$$

In fact, of all the $L^p(\mathbb{R})$ spaces, $L^2(\mathbb{R})$ is the **only one** that is a Hilbert space.

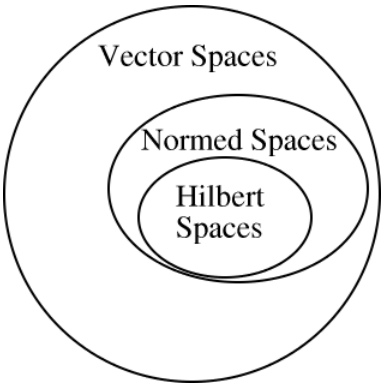


Figure 15.7.2

Hilbert spaces are by far the nicest. If you use or study orthonormal basis expansion (Section 15.9) then you will start to see why this is true.

This page titled 15.7: Common Hilbert Spaces is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.8: Types of Bases

Normalized Basis

Definition: Normalized Basis

A basis $\{b_i\}$ where each b_i has unit norm

$$\|b_i\| = 1, \quad i \in \mathbb{Z}$$

Note

The concept of basis applies to all vector spaces (Section 15.2). The concept of **normalized basis** applies only to normed spaces (Section 15.3).

You can always normalize a basis: just multiply each basis vector by a constant, such as $\frac{1}{\|b_i\|}$

Example 15.8.1

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:

$$\tilde{b}_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$\tilde{b}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Normalized with ℓ^1 norm:

$$\tilde{b}_0 = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$\tilde{b}_1 = \frac{1}{2} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Orthogonal Basis

Orthogonal Basis

a basis $\{bi\}$b i in which the elements are **mutually orthogonal**

$\forall i, i \neq j : (\langle bi, bj \rangle = 0)$i i j b i b j 0

Definition: Orthogonal Basis

A basis $\{b_i\}$ in which the elements are **mutually orthogonal**

$$\langle b_i, b_j \rangle = 0, \quad i \neq j$$

Note

The concept of orthogonal basis applies only to Hilbert Spaces (Section 15.4).

Example 15.8.2

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 15.8.3

Now we have the following basis and relationship:

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\} = \{h_0, h_1\}$$

$$\langle h_0, h_1 \rangle = 1 \times 1 + 1 \times -1 = 0$$

Orthonormal Basis

Pulling the previous two sections (definitions) together, we arrive at the most important and useful basis type:

Definition: Orthonormal Basis

A basis that is both **normalized** and **orthogonal**

$$\|b_i\| = 1, \quad i \in \mathbb{Z}$$

$$\langle b_i, b_j \rangle \quad , \quad i \neq j$$

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 1.6) and is also often written as $\delta[i-j]$.

Orthonormal Basis Example #1

$$\{b_0, b_2\} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

Orthonormal Basis Example #2

$$\{b_0, b_2\} = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

Orthonormal Basis Example #3

$$\{b_0, b_2\} = \left\{ \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

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

$$x = \sum_i \alpha_i b_i$$

It is easy to find the α_i:

$$\begin{aligned} \langle x, b_i \rangle &= \left\langle \sum_k \alpha_k b_k, b_i \right\rangle \\ &= \sum_k \alpha_k \langle (b_k, b_i) \rangle \end{aligned} \tag{15.8.1}$$

where in the above equation we can use our knowledge of the delta function to reduce this equation:

$$\langle b_k, b_i \rangle = \delta_{ik} = \begin{cases} 1 \text{ if } i = k \\ 0 \text{ if } i \neq k \end{cases}$$
$$\langle x, b_i \rangle = \alpha_i$$

Therefore, we can conclude the following important equation for x:

$$x = \sum_i \langle (x, b_i) \rangle \, b_i$$

The α_i's are easy to compute (no interaction between the b_i's)

Example 15.8.4

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 15.8.5: Slightly Modified Fourier Series

We are given the basis

$$\left\{ \frac{1}{\sqrt{T}} e^{j\omega_0 nt} \right\} \Big|_{n=-\infty}^{\infty}$$

on $L^2([0, T])$ where $T = \frac{2\pi}{\omega_0}$.

$$f(t) = \sum_{n=-\infty}^{\infty} \langle (f, e^{j\omega_0 nt}) \rangle \, e^{j\omega_0 nt} \frac{1}{\sqrt{T}}$$

Where we can calculate the above inner product in L^2 as

$$\langle f, e^{j\omega_0 nt} \rangle = \frac{1}{\sqrt{T}} \int_0^T f(t) \overline{e^{j\omega_0 nt}} \mathrm{d}t = \frac{1}{\sqrt{T}} \int_0^T f(t) e^{-(j\omega_0 nt)} \mathrm{d}t$$

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

$$x = \sum_i \alpha_i b_i$$

where $\alpha_i = \langle x, b_i \rangle$.

- "Analysis": decomposing x in term of the b_i

$$\alpha_i = \langle x, b_i \rangle$$

- "Synthesis": building x up out of a weighted combination of the b_i

$$x = \sum_i \alpha_i b_i$$

This page titled 15.8: Types of Bases is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.9: Orthonormal Basis Expansions

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 14.2) 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 14.5).

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.

Example 15.9.1

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 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.

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 S. That is, we can find $\{\alpha_i\}$, where $\alpha_i \in \mathbb{C}$ (scalars) such that

$$\boldsymbol{x} = \sum_i \alpha_i b_i, x \in S$$

where $\boldsymbol{x}$ is a vector in S, α is a scalar in $\mathbb{C}$, and $\boldsymbol{b}$ 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 $\boldsymbol{x}$.

Example 15.9.2

Let us look at simple example in $\mathbb{R}^2$, where we have the following vector:

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

Standard Basis: $\{e_0, e_1\} = \{(1,0)^T, (0,1)^T\}$

$$\boldsymbol{x} = e_0 + 2e_1$$

Alternate Basis: $\{h_0, h_1\} = \{(1,1)^T, (1,-1)^T\}$

$$\boldsymbol{x} = \frac{3}{2}h_0 + \frac{-1}{2}h_1$$

In general, given a basis $\{b_0, b_1\}$ and a vector $\boldsymbol{x} \in \mathbb{R}^2$, how do we find the α_0 and α_1 such that

$$\boldsymbol{x} = \alpha_0 b_0 + \alpha_1 b_1 \tag{15.9.1}$$

Finding the Coefficients

Now let us address the question posed above about finding α_i's in general for $\mathbb{R}^2$. We start by rewriting Equation 15.9.1 so that we can stack our b_i's as columns in a 2×2 matrix.

$$(x) = \alpha_0 (b_0) + \alpha_1 (b_1) \tag{15.9.2}$$

$$(x) = \begin{pmatrix} \vdots & \vdots \\ b_0 & b_1 \\ \vdots & \vdots \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \tag{15.9.3}$$

Example 15.9.3

Here is a simple example, which shows a little more detail about the above equations.

$$\begin{aligned} \begin{pmatrix} x[0] \\ x[1] \end{pmatrix} &= \alpha_0 \begin{pmatrix} b_0[0] \\ b_0[1] \end{pmatrix} + \alpha_1 \begin{pmatrix} b_1[0] \\ b_1[1] \end{pmatrix} \\ &= \begin{pmatrix} \alpha_0 b_0[0] + \alpha_1 b_1[0] \\ \alpha_0 b_0[1] + \alpha_1 b_1[1] \end{pmatrix} \end{aligned} \tag{15.9.4}$$

$$\begin{pmatrix} x[0] \\ x[1] \end{pmatrix} = \begin{pmatrix} b_0[0] & b_1[0] \\ b_0[1] & b_1[1] \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$$

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**:

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$$

This gives us the following, concise equation:

$$\boldsymbol{x} = B\boldsymbol{\alpha} \tag{15.9.5}$$

which is equivalent to $\boldsymbol{x} = \sum_{i=0}^{1} \alpha_i b_i$.

Example 15.9.4

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_is, we solve for the coefficient vector in Equation 15.9.5

$$\boldsymbol{\alpha} = B^{-1}\boldsymbol{x} \tag{15.9.6}$$

Where B^{-1} is the inverse matrix of B.

Examples

Example 15.9.5

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,

$$\alpha = B^{-1}x = x$$

Example 15.9.6

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

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

Now we solve for α

$$\alpha = B^{-1}x = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{-1}{2} \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 2.5 \\ 0.5 \end{pmatrix}$$

and we get

$$x = 2.5h_0 + 0.5h_1$$

Exercise 15.9.1

Now we are given the following basis matrix and x:

$$\{b_0, b_1\} = \left\{ \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix} \right\}$$
$$x = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

For this problem, make a sketch of the bases and then represent x in terms of b_0 and b_1.

Answer

In order to represent x in terms of b_0 and b_1 we will follow the same steps we used in the above example.

$$B = \begin{pmatrix} 1 & 2 \\ 3 & 0 \end{pmatrix}$$

$$B^{-1} = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{-1}{6} \end{pmatrix}$$

$$\boldsymbol{\alpha} = B^{-1}\boldsymbol{x} = \begin{pmatrix} 1 \\ \frac{2}{3} \end{pmatrix}$$

And now we can write $\boldsymbol{x}$ in terms of b_0 and b_1.

$$\boldsymbol{x} = b_0 + \frac{2}{3}b_1$$

And we can easily substitute in our known values of b_0 and b_1 to verify our results.

Note

A change of basis simply looks at $\boldsymbol{x}$ from a "different perspective." B^{-1} **transforms** $\boldsymbol{x}$ from the standard basis to our new basis, $\{b_0, b_1\}$. Notice that this is a totally mechanical procedure.

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 $\{b_0, b_1, \ldots, b_{n-1}\}$ for $\mathbb{R}^n$, we want to find $\{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ such that

$$\boldsymbol{x} = \alpha_0 b_0 + \alpha_1 b_1 + \ldots + \alpha_{n-1} b_{n-1} \tag{15.9.7}$$

Again, we will set up a basis matrix

$$B = \begin{pmatrix} b_0 & b_1 & b_2 & \ldots & b_{n-1} \end{pmatrix}$$

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 Equation 15.9.5

$$\boldsymbol{x} = \begin{pmatrix} b_0 & b_1 & \ldots & b_{n-1} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{n-1} \end{pmatrix} = B\boldsymbol{\alpha}$$

and

$$\boldsymbol{\alpha} = B^{-1}\boldsymbol{x}$$

This page titled 15.9: Orthonormal Basis Expansions is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.10: Function Space

We can also find basis vectors (Section 15.9) for vector spaces (Section 15.2) 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 15.10.1

$P_2 = \{$ 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.

$$f(t) = t^2 - 3t - 4$$

Alternate basis

$$\{b_0(t), b_1(t), b_2(t)\} = \left\{1, t, \frac{1}{2}\left(3t^2 - 1\right)\right\}$$

write $f(t)$ in terms of this new basis $d_0(t) = b_0(t)$, $d_1(t) = b_1(t)$, $d_2(t) = \frac{3}{2}b_2(t) - \frac{1}{2}b_0(t)$.

$$f(t) = t^2 - 3t - 4 = 4b_0(t) - 3b_1(t) + b_2(t)$$
$$f(t) = \beta_0 d_0(t) + \beta_1 d_1(t) + \beta_2 d_2(t) = \beta_0 b_0(t) + \beta_1 b_1(t) + \beta_2\left(\tfrac{3}{2}b_2(t) - \tfrac{1}{2}b_0(t)\right)$$
$$f(t) = \beta_0 b_0(t) + \beta_1 b_1(t) + \tfrac{3}{2}\beta_2 b_2(t)$$

so

$$\beta_0 - \tfrac{1}{2} = 4$$
$$\beta_1 = -3$$
$$\tfrac{3}{2}\beta_2 = 1$$

then we get

$$f(t) = 4.5d_0(t) - 3d_1(t) + \frac{2}{3}d_2(t)$$

Example 15.10.2

$e^{j\omega_0 nt}\big|_{n=-\infty}^{\infty}$ 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:

$$C_n = \frac{1}{T}\int_0^T \left(f(t)e^{-(j\omega_0 nt)}\right)dt$$

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).

This page titled 15.10: Function Space is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.11: Haar Wavelet Basis

Introduction

Fourier series is a useful orthonormal representation (Section 15.9) 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 6.7)).

Wavelets, discovered in the last 15 years, are another kind of basis for $L^2([0,T])$ and have many nice properties.

Basis Comparisons

Fourier series - c_n give frequency information. Basis functions last the entire interval.

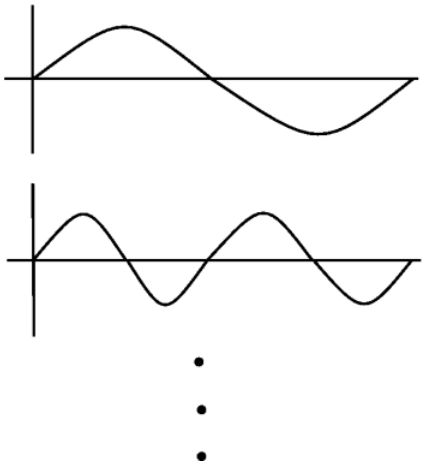


Figure 15.11.1: Fourier basis functions

Wavelets - basis functions give frequency info but are **local** in time.

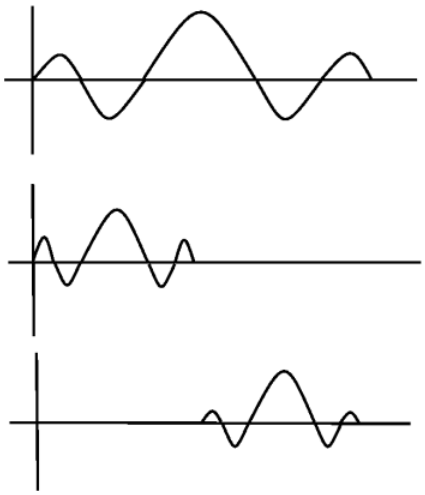


Figure 15.11.2: Wavelet basis functions

In Fourier basis, the basis functions are **harmonic multiples** of $e^{j\omega_0 t}$

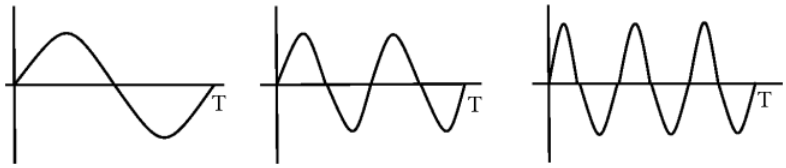


Figure 15.11.3: basis $= \left\{\frac{1}{\sqrt{T}} e^{j\omega_0 nt}\right\}$

In Haar wavelet basis, the basis functions are **scaled and translated** versions of a "mother wavelet" $\psi(t)$.

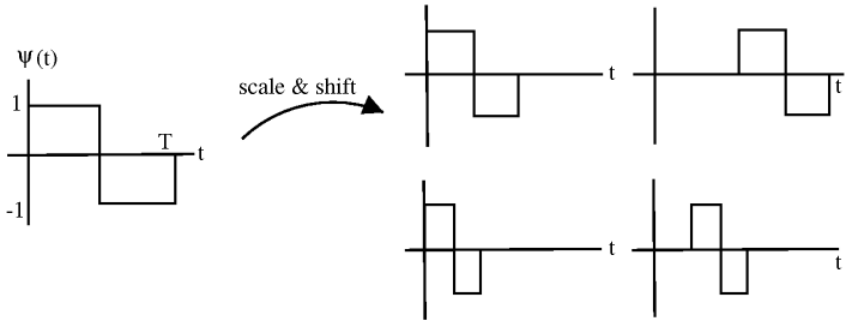


Figure 15.11.4

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 $\left\{\phi(t), 2^{\frac{j}{2}}\psi\left(2^j t - k\right), \phi(t), 2^{\frac{j}{2}}\psi\left(2^j t - k\right) \mid j \in \mathbb{Z} \text{ and } (k = 0, 1, 2, \ldots, 2^j - 1)\right\}$.

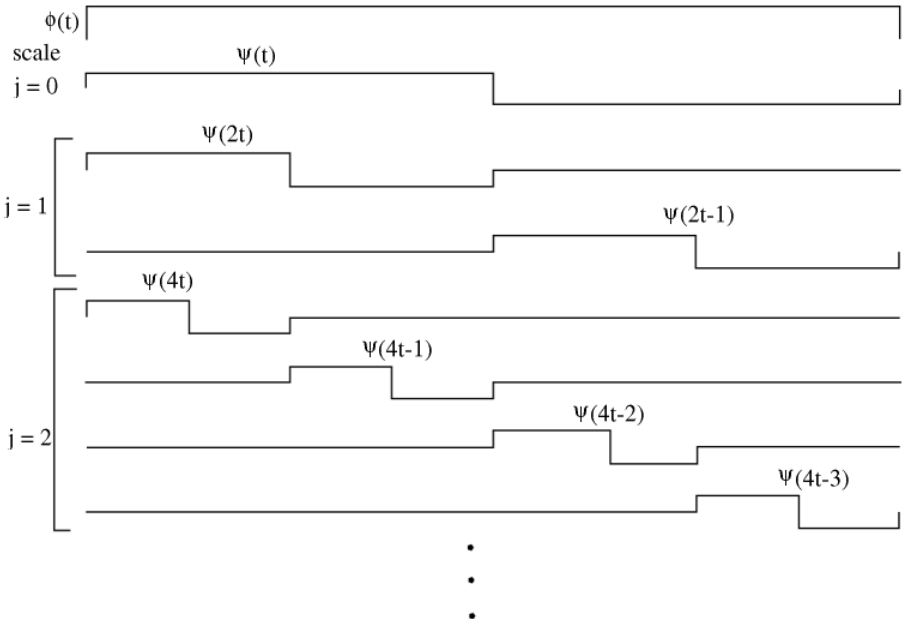


Figure 15.11.5

$$\psi(t) = \begin{cases} 1 \text{ if } 0 \leq t < \frac{T}{2} \\ -1 \text{ if } 0 \leq \frac{T}{2} < T \end{cases}$$

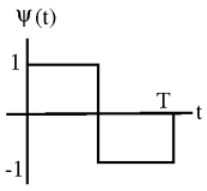


Figure 15.11.6

Let $\psi_{j,k}(t) = 2^{\frac{j}{2}}\psi\left(2^j t - k\right)$.

https://eng.libretexts.org/@go/page/23201

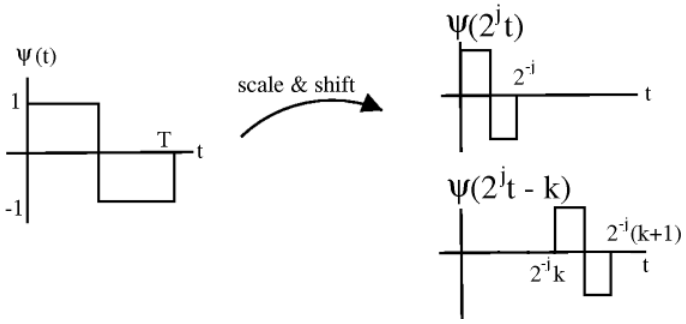


Figure 15.11.7

Larger $j \rightarrow$ "skinnier" basis function, $j = \{0, 1, 2, ...\}$, 2^j shifts at each scale: $k = 0, 1, ..., 2^j - 1$

Check: each $\psi_{j,k}(t)$ has unit energy

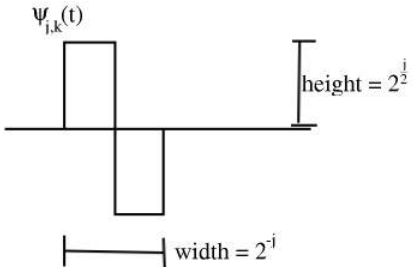
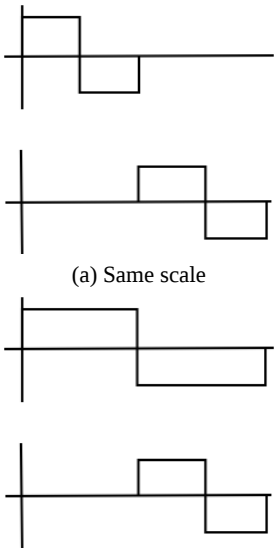


Figure 15.11.8

$$\left(\int \psi_{j,k}^2(t)\mathrm{d}t = 1\right) \Rightarrow \left(\|\psi_{j,k}(t)\|_2 = 1\right)$$

Any two basis functions are orthogonal.



(a) Same scale

(b) Different Scale

Figure 15.11.9: Integral of product = 0

Also, $\{\psi_{j,k}, \phi\}$ span $L^2([0, T])$.

Haar Wavelet Transform

Using what we know about Hilbert spaces (Section 15.4): For any $f(t) \in L^2([0,T])$, we can write

Synthesis

$$f(t) = \sum_j \sum_k w_{j,k}\psi_{j,k}(t) + c_0\phi(t)$$

Analysis

$$w_{j,k} = \int_0^T f(t)\psi_{j,k}(t)dt$$

$$c_0 = \int_0^T f(t)\phi(t)dt$$

> Note
>
> the $w_{j,k}$ are **real**

The Haar transform is **super useful** especially in **image compression**

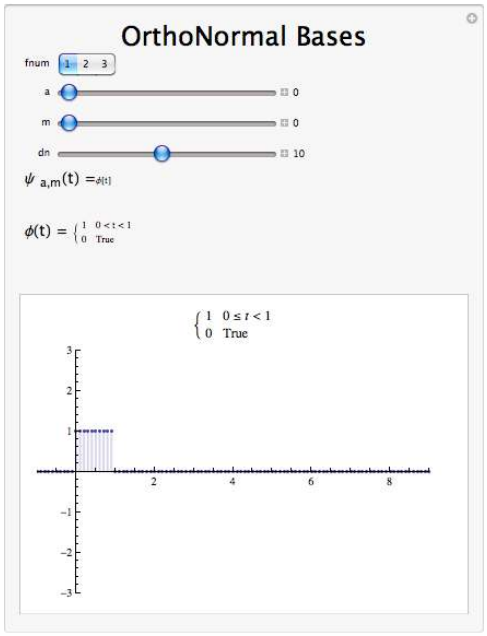


Figure 15.11.10: Interact (when online) with a Mathematica CDF demonstrating the Haar Wavelet as an Orthonormal Basis.

This page titled 15.11: Haar Wavelet Basis is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.12: Orthonormal Bases in Real and Complex Spaces

Notation

Transpose operator A^T flips the matrix across it's diagonal.

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$
$$A^T = \begin{pmatrix} a_{1,1} & a_{2,1} \\ a_{1,2} & a_{2,2} \end{pmatrix}$$

Column i of A is **row** i of A^T

Recall, inner product

$$\boldsymbol{x} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

$$\boldsymbol{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

$$\boldsymbol{x}^T\boldsymbol{y} = \begin{pmatrix} x_0 & x_1 & \dots & x_{n-1} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \sum_i \boldsymbol{x}_i \boldsymbol{y}_i = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$$

on $\mathbb{R}^n$.

Hermitian transpose A^H, transpose and conjugate

$$A^{\mathrm{H}} = \overline{A^T}$$
$$\langle \boldsymbol{y}, \boldsymbol{x} \rangle = \boldsymbol{x}^{\mathrm{H}}\boldsymbol{y} = \textstyle\sum_i \boldsymbol{x}_i \bar{\boldsymbol{y}}_i$$

on $\mathbb{C}^n$.

Now, let $\{b_0, b_1, \dots, b_{n-1}\}$ be an orthonormal basis for $\mathbb{C}^n$

$$i = \{0, 1, \dots, n-1\} \, \langle b_i, b_i \rangle = 1$$
$$\left(i \neq j, \langle b_i, b_i \rangle = b_j^H b_i = 0\right)$$

Basis matrix:

$$B = \begin{pmatrix} \vdots & \vdots & & \vdots \\ b_0 & b_1 & \dots & b_{n-1} \\ \vdots & \vdots & & \vdots \end{pmatrix}$$

Now,

$$B^{\mathrm{H}}B = \begin{pmatrix} \cdots & b_0^{\mathrm{H}} & \cdots \\ \cdots & b_1^{\mathrm{H}} & \cdots \\ & \vdots & \\ \cdots & b_{n-1}^{\mathrm{H}} & \cdots \end{pmatrix} \begin{pmatrix} \vdots & \vdots & & \vdots \\ b_0 & b_1 & \cdots & b_{n-1} \\ \vdots & \vdots & & \vdots \end{pmatrix} = \begin{pmatrix} b_0^{\mathrm{H}}b_0 & b_0^{\mathrm{H}}b_1 & \dots & b_0^{\mathrm{H}}b_{n-1} \\ b_1^{\mathrm{H}}b_0 & b_1^{\mathrm{H}}b_1 & \dots & b_1^{\mathrm{H}}b_{n-1} \\ \vdots & & & \\ b_{n-1}^{\mathrm{H}}b_0 & b_{n-1}^{\mathrm{H}}b_1 & \dots & b_{n-1}^{\mathrm{H}}b_{n-1} \end{pmatrix}$$

For orthonormal basis with basis matrix B

$$B^H = B^{-1}$$

($B^T = B^{-1}$in $\mathbb{R}^n$ 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

$$\boldsymbol{x} = \sum_i \alpha_i b_i$$

Calculate

$$(\boldsymbol{\alpha} = B^{-1}\boldsymbol{x}) \Rightarrow (\boldsymbol{\alpha} = B^H \boldsymbol{x})$$

Using an orthonormal basis we rid ourselves of the inverse operation.

This page titled 15.12: Orthonormal Bases in Real and Complex Spaces is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.13: Plancharel and Parseval's Theorems

Parseval's Theorem

Continuous Time Fourier Series preserves signal energy

i.e.:

$$\int_0^T |f(t)|^2 dt = T \sum_{n=-\infty}^{\infty} |C_n|^2 \quad \text{with unnormalized basis } e^{j\frac{2\pi}{T}nt}$$

$$\int_0^T |f(t)|^2 dt = \sum_{n=-\infty}^{\infty} |C_n|^2 \quad \text{with unnormalized basis } \frac{e^{j\frac{2\pi}{T}nt}}{\sqrt{T}}$$

$$\underbrace{\|f\|_2^2}_{L^2[0,T) energy} = \underbrace{\|C_n'\|_2^2}_{l^2(Z) energy}$$

Prove: Plancherel theorem

$$\text{Given } f(t) \overset{\text{CTFS}}{\longrightarrow} c_n$$
$$g(t) \overset{\text{CTFS}}{\longrightarrow} d_n$$

$$\text{Then } \int_0^T f(t)g^*(t)dt = T \sum_{n=-\infty}^{\infty} c_n d_n^* \text{ with unnormalized basis } e^{j\frac{2\pi}{T}nt}$$

$$\int_0^T f(t)g^*(t)dt = \sum_{n=-\infty}^{\infty} c_n' \left(d_n'\right)^* \text{ with normalized basis } \frac{e^{j\frac{2\pi}{T}nt}}{\sqrt{T}}$$

$$\langle f, g \rangle_{L_2(0,T]} = \langle c, d \rangle_{l_2(\mathbb{Z})}$$

Periodic Signals Power

$$\text{Energy } = \|f\|^2 = \int_{-\infty}^{\infty} |f(t)|^2 dt = \infty$$

$$\begin{aligned} \text{Power } &= \lim_{T\to\infty} \frac{\text{Energy in } [0,T)}{T} \\ &= \lim_{T\to\infty} \frac{T\sum_n |c_n|^2}{T} \\ &= \sum_{n\in\mathbb{Z}} |c_n|^2 \text{ (unnormalized FS)} \end{aligned}$$

Example 15.13.1: Fourier Series of Square Pulse III - Compute the Energy

Figure 15.13.1

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi}{T}nt} \overset{\mathbb{FS}}{\to} c_n = \frac{1}{2}\frac{\sin\frac{\pi}{2}n}{\frac{\pi}{2}n}$$

$$\text{energy in time domain: } \|f\|_2^2 = \int_0^T |f(t)|^2 dt = \frac{T}{2}$$

Apply Parseval's Theorem:

$$T\sum_n |c_n|^2$$

$$= \frac{T}{4} \sum_n \left(\frac{\sin \frac{\pi}{2} n}{\frac{\pi}{2} n} \right)^2$$
$$= \frac{T}{4} \frac{4}{\pi^2} \sum_n \frac{\left(\sin \frac{\pi}{2} n\right)^2}{n^2}$$
$$= \frac{T}{\pi^2} \left[\frac{\pi^2}{4} + \underbrace{\sum_n \text{odd} \frac{1}{n^2}}_{\frac{\pi^2}{4}} \right]$$
$$= \frac{T}{2} \square$$

Plancharel Theorem

Theorem 15.13.1: 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. $x \in H$, $y \in H$

$$x = \sum_i \alpha_i b_i$$
$$y = \sum_i \beta_i b_i$$

then

$$\langle x, y \rangle_H = \sum_i \alpha_i \overline{\beta_i}$$

Example 15.13.2

Applying the Fourier Series, we can go from $f(t)$ to $\{c_n\}$ and $g(t)$ to $\{d_n\}$

$$\int_0^T f(t) \overline{g(t)} \mathrm{d}t = \sum_{n=-\infty}^{\infty} c_n \overline{d_n}$$

inner product in time-domain = inner product of Fourier coefficients.

Proof:

$$x = \sum_i \alpha_i b_i$$
$$y = \sum_j \beta_j b_j$$

$$\langle x, y \rangle_H = \left\langle \sum_i \alpha_i b_i, \sum_j \beta_j b_j \right\rangle = \sum_i \alpha_i \left\langle \left(b_i, \sum_j \beta_j b_j \right) \right\rangle = \sum_i \alpha_i \sum_j \bar{\beta}_j \langle (b_i, b_j) \rangle = \sum_i \alpha_i \bar{\beta}_i$$

by using inner product rules (Section 15.4)

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

Plancharels Theorem Demonstration



Figure 15.13.2: Interact (when online) with a Mathematica CDF demonstrating Plancharels Theorem visually. To Download, right-click and save target as .cdf.

Parseval's Theorem: a different approach

Theorem 15.13.2: Parseval's Theorem

Energy of a signal = sum of squares of its expansion coefficients

Let $x \in H$, $\{b_i\}$ ONB

$$x = \sum_i \alpha_i b_i$$

Then

$$(\|x\|_H)^2 = \sum_i (|\alpha_i|)^2$$

Proof:

Directly from Plancharel

$$(\|x\|_H)^2 = \langle x, x \rangle_H = \sum_i \alpha_i \overline{\alpha_i} = \sum_i (|\alpha_i|)^2$$

Example 15.13.3

Fourier Series $\frac{1}{\sqrt{T}} e^{jw_0 nt}$

$$f(t) = \frac{1}{\sqrt{T}} \sum_n c_n \frac{1}{\sqrt{T}} e^{jw_0 nt}$$
$$\int_0^T (|f(t)|)^2 dt = \sum_{n=-\infty}^{\infty} (|c_n|)^2$$

This page titled 15.13: Plancharel and Parseval's Theorems is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

15.14: Approximation and Projections in Hilbert Space

Introduction

Given a line 'l' and a point 'p' in the plane, what's the closest point 'm' to 'p' on 'l'?

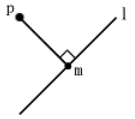


Figure 15.14.1: Figure of point 'p' and line 'l' mentioned above.

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 span{v} **best approximates** x?)

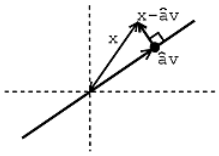


Figure 15.14.2

The condition is that $x - \widehat{\alpha} v$ and αv are **orthogonal**.

Calculating α

How to calculate $\hat{a}$?

We know that $(x - \widehat{\alpha} v)$ is perpendicular to every vector in span{v}, so

$$\langle x - \widehat{\alpha} v, \beta v \rangle = 0, \forall(\beta)$$
$$\bar{\beta}\langle (x, v) \rangle - \widehat{\alpha}\bar{\beta}\langle (v, v) \rangle = 0$$

because $\langle v, v \rangle = 1$, so

$$(\langle (x, v) \rangle - \hat{\alpha} = 0) \Rightarrow (\hat{\alpha} = \langle x, v \rangle)$$

Closest vector in span{v} = $\langle (x, v) \rangle v$, where $\langle (x, v) \rangle v$ is the projection of x onto v.

We can do the same thing in higher dimensions.

Exercise 15.14.1

Let $V \subset H$ be a subspace of a Hilbert space (Section 15.4) H. Let $x \in H$ be given. Find the $y \in V$ that **best approximates** x. i.e., $\|x - y\|$ is minimized.

Answer

1. Find an orthonormal basis $\{b_1, \ldots, b_k\}$ 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 $(x - y) \perp V (\langle x - y, v \rangle = 0, \quad \forall(v) \in V$

Example 15.14.1

$x \in \mathbb{R}^3$, $V = \text{span}\left(\left\{\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$, $x = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$. 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 15.14.2

V = {space of periodic signals with frequency no greater than $3w_0$}. Given periodic f(t), what is the signal in V that best approximates f?

1. $\left\{ \frac{1}{\sqrt{T}} e^{jw_0 kt}, k = -3, -2, \dots, 2, 3 \right\}$ is an ONB for V
2. $g(t) = \frac{1}{T} \sum_{k=-3}^{3} \langle (f(t), e^{jw_0 kt}) \rangle \, e^{jw_0 kt}$ is the closest signal in V to f(t) ⇒ reconstruct f(t) using only 7 terms of its Fourier series.

Example 15.14.3

Let V = {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?

$$g(t) = \sum_{i=-\infty}^{\infty} \langle (f, b_i) \rangle \, b_i$$
$$\langle f, b_i \rangle = \int_{-\infty}^{\infty} f(t) b_i(t) \mathrm{d}t = \int_{i-1}^{i} f(t) \mathrm{d}t$$

Example 15.14.4

This demonstration explores approximation using a Fourier basis and a Haar Wavelet basis. See here for instructions on how to use the demo.

This page titled 15.14: Approximation and Projections in Hilbert Space is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

16: Appendix C- Analysis Topics Overview

16.1: Introduction to Metric Spaces

16.2: Convergence of Sequences

16.3: Convergence of Sequences of Vectors

16.4: Uniform Convergence of Function Sequences

This page titled 16: Appendix C- Analysis Topics Overview is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

16.1: Introduction to Metric Spaces

Introduction

In may courses, concepts such as continuity and convergence are invoked without much discussion of their formal definitions, instead relying on the reader's intuitive understanding of these matters. However, for purposes of proofs, including some in the main body of material for this course, a greater rigor is required. This module will discuss metric spaces, a mathematical construct that provide a framework for the study continuity, convergence, and other related ideas in their most concrete but still formal senses. This is accomplished by formalizing a notion of the distance between two elements in a set. The intent in this and subsequent modules in this chapter is not to give a complete overview of the basic topics of analysis but to give a short introduction to those most important to discussion of signal processing in this course.

Metric Spaces

A Notion of Distance

In many situations in signal processing it is often useful to have a concept of distance between the points in a set. This notion is mathematically formalized through the idea of a metric space. A metric space (M, d) is a set M together with a function $d : M \times M \rightarrow R$ that assigns distances between pairs of elements in M while satisfying three conditions. First, for every $x, y \in M, d(x, y) \geq 0$ with $d(x, y) = 0$ if and only if $x = y$. Second, for every $x, y \in M, d(x, y) = d(y, x)$ symmetrically. Third, for every $x, y, z \in M, d(x, y) + d(x, z) \geq d(y, z)$, which is known as the triangle inequality.

There are, of course, several different possible choices of definitions for distances in a given set. Our typical intuitive understanding of distance in $\mathbb{R}^n$ fits within this framework as the standard Euclidean metric

$$d(x, y) = \|x - y\|_2$$

as does the taxicab or Manhatten metric

$$d(x, y) = \|x - y\|_1$$

that sums individual components of vectors, representing, for example, distances traveled walking around city blocks. Another simple yet more exotic example is provided by the discrete metric on any set defined by

$$d(x, y) = \begin{cases} 0 & x = y \\ 1 & x \neq y \end{cases}$$

in which all pairs of distinct points are equidistant from each other but every point is distance zero from itself. One can check that these satisfy the conditions for metric spaces.

Relationship with Norms

It is not surprising that norms, which provide a notion of size, and metrics, which provide a notion of distance, would have a close relationship. Intuitively, one way of defining the distance between two points in a metric space could be the size of their difference. In other words given a vector space V over the field F with norm $\| \cdot \|$, we might ask if the function

$$d(x, y) = \|x - y\|$$

for every $x, y \in V$ satisfies the conditions for (V, d) to be a metric space.

Let V be a vector space over the field F with norm $\| \cdot \|$, and let $d(x, y) = \|x - y\|$. Recall that since $\| \cdot \|$ is a norm, $||x|| = 0$ if and only if $x = 0$ and $||ax|| = |a|||x||$ for all $a \in F$ and $x \in V$. Hence $||x - y|| \geq 0$ for all $x, y \in V$ and $\|x - y\| = 0$ if and only if $x = y$. Since $y - x = -(x - y)$ and $\| - (x - y)\| = \|x - y\|$ it follows that $||x - y|| = ||y - x \mid$ for all $x, y \in V$. Finally, $\|x\| + \|y\| \geq \|x + y\|$ by the properties of norms, so $\|x - y\| + \|x - z\| \geq \|y - z\|$ for all $x, y, z \in V$. Thus, (V, d) does indeed satisfy the conditions to be a metric space and is discussed as the metric space induced by the norm $\| \cdot \|$.

Metric Spaces Summary

Metric spaces provide a notion of distance and a framework with which to formally study mathematical concepts such as continuity and convergence, and other related ideas. Many metrics can be chosen for a given set, and our most common notions of distance satisfy the conditions to be a metric. Any norm on a vector space induces a metric on that vector space and it is in these types of metric spaces that we are often most interested for study of signals and systems.

This page titled 16.1: Introduction to Metric Spaces is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

16.2: Convergence of Sequences

Sequences

Definition: 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 16.2.1

A real number sequence:

$$g_n = \frac{1}{n}$$

Example 16.2.2

A vector sequence:

$$g_n = \begin{pmatrix} \sin\left(\frac{n\pi}{2}\right) \\ \cos\left(\frac{n\pi}{2}\right) \end{pmatrix}$$

Example 16.2.3

A function sequence:

$$g_n(t) = \begin{cases} 1 & \text{if } 0 \le t < \frac{1}{n} \\ 0 & \text{otherwise} \end{cases}$$

Note

A function can be thought of as an infinite dimensional vector where for each value of 't' we have one dimension

Convergence of Real Sequences

Definition: Limit

A sequnce $\{g_n\}|_{n=1}^{\infty}$ converges to a limit $g \in \mathbb{R}$ if for **every** $\varepsilon > 0$ there is an integer N such that

$$|g_i - g| < \varepsilon, \quad i \ge N$$

We usually denote a limit by writing

$$\text{limit}_{i\to\infty}\, g_i = g$$

or

$$g_i \to g$$

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 16.2.4

We are given the following convergent sequence:

$$g_n = \frac{1}{n}$$

Intuitively we can assume the following limit:

$$\text{limit}_{n\to\infty}\, g_n = 0$$

Let us prove this rigorously. Say that we are given a real number $\varepsilon > 0$. Let us choose $N = \lceil \frac{1}{\varepsilon} \rceil$, where $[x]$ denotes the smallest integer larger than x. Then for $n \geq N$ we have

$$|g_n - 0| = \frac{1}{n} \leq \frac{1}{N} < \varepsilon$$

Thus,

$$\text{limit}_{n\to\infty}\, g_n = 0$$

Example 16.2.5

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.

This page titled 16.2: Convergence of Sequences is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

16.3: Convergence of Sequences of Vectors

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 16.4), can also be studied. For this discussion , we will assume that the vectors belong to a normed vector space (Section 15.3).

Pointwise Convergence

A sequence (Section 16.2) $\{g_n\}|_{n=1}^{\infty}$ converges **pointwise** to the limit g if each element of g_n converges to the corresponding element in g. Below are few examples to try and help illustrate this idea.

Example 16.3.1

$$g_n = \begin{pmatrix} g_n[1] \\ g_n[2] \end{pmatrix} = \begin{pmatrix} 1+\frac{1}{n} \\ 2-\frac{1}{n} \end{pmatrix}$$

First we find the following limits for our two g_n's:

$$\begin{aligned} \text{limit}_{n\to\infty}\, g_n[1] &= 1 \\ \text{limit}_{n\to\infty}\, g_n[2] &= 2 \end{aligned}$$

Therefore we have the following,

$$\text{limit}_{n\to\infty}\, g_n = \boldsymbol{g}$$

pointwise, where $\boldsymbol{g} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Example 16.3.2

$$g_n(t) = \frac{t}{n}, t \in \mathbb{R}$$

As done above, we first want to examine the limit

$$\text{limit}_{n\to\infty}\, g_n\,(t_0) = \text{limit}_{n\to\infty}\, \frac{t_0}{n} = 0$$

where $t_0 \in \mathbb{R}$. Thus $\text{limit}_{n\to\infty}\, g_n = g$ pointwise where $g(t) = 0$ for all $t \in \mathbb{R}$.

Norm Convergence

The sequence (Section 16.2) $\{g_n\}|_{n=1}^{\infty}$ converges to g in norm if $\text{limit}_{n\to\infty} \|g_n - g\| = 0$. Here $\|\cdot\|$ is the norm $Section 15.3) of the corresponding vector space of \(g_n$'s. Intuitively this means the distance between vectors g_n and g decreases to 0.

Example 16.3.3

$$g_n = \begin{pmatrix} 1+\frac{1}{n} \\ 2-\frac{1}{n} \end{pmatrix}$$

Let $\boldsymbol{g} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

$$\begin{aligned} \|g_n - \boldsymbol{g}\| &= \sqrt{\left(1+\frac{1}{n}-1\right)^2 + \left(2-\frac{1}{n}\right)^2} \\ &= \sqrt{\frac{1}{n^2}+\frac{1}{n^2}} \\ &= \frac{\sqrt{2}}{n} \end{aligned}$$

Thus $\text{limit}_{n\to\infty}\|g_n - \boldsymbol{g}\| = 0$. Therefore $g_n \to \boldsymbol{g}$ in norm.

Example 16.3.4

$$g_n(t) = \begin{cases} \frac{t}{n} & \text{if } 0 \le t \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Let $g(t) = 0$ for all t.

$$\begin{aligned} \|g_n(t) - g(t)\| &= \int_0^1 \frac{t^2}{n^2} \mathrm{d}t \\ &= \left.\frac{t^3}{3n^2}\right|_{n=0}^{1} \\ &= \frac{1}{3n^2} \end{aligned}$$

Thus $\text{limit}_{n\to\infty}\|g_n(t) - g(t)\| = 0$ Therefore, $g_n(t) \to g(t)$ in norm.

Pointwise vs. Norm Convergence

Theorem 16.3.1

For $\mathbb{R}^m$, pointwise and norm convergence are equivalent.

Proof: Pointwise ⇒ Norm

$$g_n[i] \to g[i]$$

Assuming the above, then

$$\left(\|g_n - \boldsymbol{g}\|\right)^2 = \sum_{i=1}^{m} \left(g_n[i] - g[i]\right)^2$$

Thus,

$$\begin{aligned} \text{limit}_{n\to\infty}\left(\|g_n - \boldsymbol{g}\|\right)^2 &= \text{limit}_{n\to\infty}\sum_{i=1}^{m} 2 \\ &= \sum_{i=1}^{m} \text{limit}_{n\to\infty}\, 2 \\ &= 0 \end{aligned}$$

Proof: Norm ⇒ Pointwise

$$\|g_n - \boldsymbol{g}\| \to 0$$

$$\begin{aligned} \text{limit}_{n\to\infty}\sum_{i=1}^{m} 2 &= \sum_{i=1}^{m} \text{limit}_{n\to\infty}\, 2 \\ &= 0 \end{aligned} \tag{16.3.1}$$

Since each term is greater than or equal zero, all 'm' terms must be zero. Thus,

$$\text{limit}_{n\to\infty}\, 2 = 0$$

for all i. Therefore,

$$g_n \to \boldsymbol{g} \quad \text{pointwise}$$

Note

In infinite dimensional spaces the above theorem is no longer true. We prove this with counter examples shown below.

Counter Examples

Example 16.3.5: Pointwise $\nRightarrow$ 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 $\lim_{n\to\infty} g_n(t) = 0$. This means that,

$$g_n(t) \to g(t)$$

where for all t $g(t) = 0$.

Now,

$$\begin{aligned} (\|g_n\|)^2 &= \int_{-\infty}^{\infty} (|g_n(t)|)^2 \, \mathrm{d}t \\ &= \int_0^{\frac{1}{n}} n^2 \mathrm{d}t \\ &= n \to \infty \end{aligned}$$

Since the function norms blow up, they cannot converge to any function with finite norm.

Example 16.3.6: Norm $\nRightarrow$ 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,

$$\|g_n - g\| = \int_0^{\frac{1}{n}} 1 \mathrm{d}t = \frac{1}{n} \to 0$$

where $g(t) = 0$ for all t. Therefore,

$$g_n \to g \quad \text{in norm}$$

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.

This page titled 16.3: Convergence of Sequences of Vectors is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

16.4: Uniform Convergence of Function Sequences

Uniform Convergence of Function Sequences

For this discussion, we will only consider functions with g_n where

$$\mathbb{R} \to \mathbb{R}$$

Definition: Uniform Convergence

The sequence (Section 16.2) $\{g_n\}|_{n=1}^{\infty}$ converges uniformly to function g if for every $\varepsilon > 0$ there is an integer N such that $n \geq N$ implies

$$|g_n(t) - g(t)| \leq \epsilon \tag{16.4.1}$$

for **all** $t \in \mathbb{R}$.

Obviously every uniformly convergent sequence is pointwise (Section 16.3) convergent. The difference between pointwise and uniform convergence is this: If $\{g_n\}$ converges pointwise to g, then for every $\varepsilon > 0$ and for every $t \in \mathbb{R}$ there is an integer N depending on ε and t such that Equation 16.4.1 holds if $n \geq N$. If $\{g_n\}$ converges uniformly to g, it is possible for each $\varepsilon > 0$ to find one integer N that will do for all $t \in \mathbb{R}$.

Example 16.4.1

$$g_n(t) = \frac{1}{n}, t \in \mathbb{R}$$

Let $\varepsilon > 0$ be given. Then choose $N = \lceil \frac{1}{\varepsilon} \rceil$. Obviously,

$$|g_n(t) - 0| \leq \epsilon, \quad n \geq N$$

for all t. Thus, $g_n(t)$ converges uniformly to 0.

Example 16.4.2

$$g_n(t) = \frac{t}{n}, t \in \mathbb{R}$$

Obviously for any $\varepsilon > 0$ we cannot find a single function $g_n(t)$ for which Equation 16.4.1 holds with $g(t) = 0$ for all t. Thus g_n is not uniformly convergent. However we do have:

$$g_n(t) \to g(t) \text{ pointwise}$$

Conclusion

Uniform convergence always implies pointwise convergence, but pointwise convergence does not guarantee uniform convergence.

This page titled 16.4: Uniform Convergence of Function Sequences is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

CHAPTER OVERVIEW

17: Appendix D- Viewing Interactive Content

17.1: Viewing Embedded LabVIEW Content in Connexions
17.2: Getting Started with Mathematica

This page titled 17: Appendix D- Viewing Interactive Content is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

17.1: Viewing Embedded LabVIEW Content in Connexions

Introduction

In order to view LabVIEW content embedded in Connexions modules, you must install and enable the LabVIEW 8.0 and 8.5 Local VI Execution Browser Plug-in for Windows. Step-by-step installation instructions are given below. Once installation is complete, the placeholder box at the bottom of this module should display a demo LabVIEW virtual instrument (VI).

Installing the LabVIEW Run-time Engine on Microsoft Windows

1. Download and install the LabVIEW 8.0 Runtime Engine found at: zone.ni.com/devzone/cda/tut/p/id/4346.
2. Download and install the LabVIEW 8.5 Runtime Engine found at: zone.ni.com/devzone/cda/tut/p/id/6633.
3. Download the `LVBrowserPlugin.ini` file from http://zone.ni.com/devzone/cda/tut/p/id/8288, and place it in the `My Documents\LabVIEW Data` folder. (You may have to create this folder if it doesn't already exist.)
4. Restart your computer to complete the installation.
5. The placeholder box at the bottom of this module should now display a demo LabVIEW virtual instrument (VI).

Example Virtual Instrument

Figure 17.1.1: Digital filter design LabVIEW virtual instrument from http://cnx.org/content/m13115/latest/.

This page titled 17.1: Viewing Embedded LabVIEW Content in Connexions is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

https://eng.libretexts.org/@go/page/22949

17.2: Getting Started with Mathematica

What is Mathematica?

Mathematica is a computational software program used in technical fields. It is developed by Wolfram Research. Mathematica makes it easy to visualize data and create GUIs in only a few lines of code.

How can I run, create, and find Mathematica files?

Run

The free CDF Player is available for running non-commercial Mathematica programs. The option exists of downloading source files and running on your computer, but the CDF-player comes with a plug-in for viewing dynamic content online on your web browser!

Create

Mathematica 8 is available for purchase from Wolfram. Many universities (including Rice) and companies already have a Mathematica license. Wolfram has a free, save-disabled 15-day trial version of Mathematica.

Find

Wolfram has thousands of Mathematica programs (including source code) available at the Wolfram Demonstrations Project. Anyone can create and submit a Demonstration. Also, many other websites (including Connexions) have a lot of Mathematica content.

What do I need to run interactive content?

Mathematica 8 is supported on Linux, Microsoft Windows, Mac OS X, and Solaris. Mathematica's free CDF-player is available for Windows and Mac OS X, and is in development for Linux; the CDF-Player plugin is available for IE, Firefox, Chrome, Safari, and Opera.

How can I upload a Mathematica file to a Connexions module?

Go to the Files tab at the top of the module and upload your .cdf file, along with an (optional) screenshot of the file in use. In order to generate a clean bracket-less screenshot, you should do the following:

- Open your .cdf in Mathematica and left click on the bracket surrounding the manipulate command.
- Click on Cell->Convert To->Bitmap.
- Then click on File->Save Selection As, and save the image file in your desired image format.

Embed the files into the module in any way you like. Some tags you may find helpful include image, figure, download, and link (if linking to an .cdf file on another website). The best method is to create an interactive figure, and include a fallback png image of the cdf file should the CDF image not render properly. See the interactive demo/image below.

Convolution Demo

```
<figure >
<media  alt="timeshiftDemo">
<image mime-type="image/png" src="Convolutiondisplay-4.cdf" thumbnail="Convolution4.0D
<object width="500" height="500" src="Convolutiondisplay-4.cdf" mime-type="applicatior
<image mime-type="application/postscript" for="pdf" src="Convolution4.0Display.png" wi
</media>
<caption>Interact (when online) with a Mathematica CDF demonstrating Convolution. To D
</figure>
```

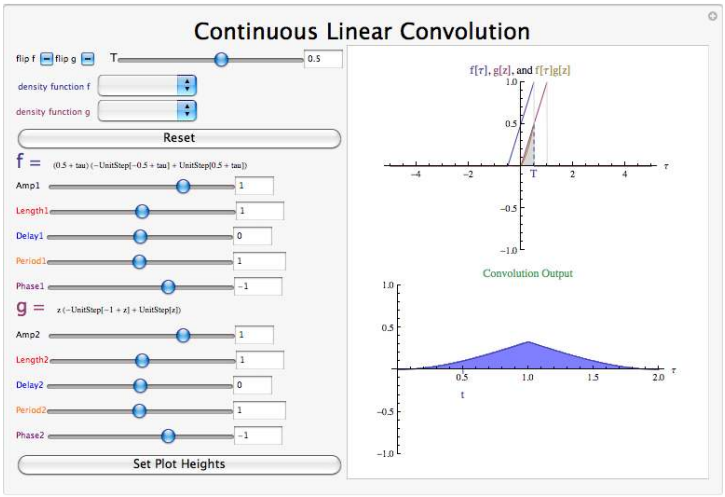


Figure 17.2.1: Interact (when online) with a Mathematica CDF demonstrating Convolution. To Download, right-click and save target as .cdf.

Alternatively, this is how it looks when you use a thumbnail link to a live online demo.

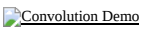


Figure 17.2.2: Click on the above thumbnail image (when online) to view an interactive Mathematica Player demonstrating Convolution.

How can I learn Mathematica?

Open Mathematica and go to the Getting Started section of the "Welcome to Mathematica" screen, or check out Help: Documentation Center.

The Mathematica Learning Center has lots of screencasts, how-tos, and tutorials.

When troubleshooting, the error messages are often unhelpful, so it's best to evaluate often so the problem can be easily located. Search engines like Google are useful when you're looking for an explanation of specific error messages.

This page titled 17.2: Getting Started with Mathematica is shared under a CC BY 3.0 license and was authored, remixed, and/or curated by Richard Baraniuk et al. via source content that was edited to the style and standards of the LibreTexts platform.

Index

A

accessibleeducation
- InfoPage

action
- 1.7: Discrete Time Impulse Function

adaptivehomework
- InfoPage

ADC
- 10.7: Discrete Time Processing of Continuous Time Signals

algorithm
- 13: Capstone Signal Processing Topics
- 13.2: The Fast Fourier Transform (FFT)
- 13.3: Deriving the Fast Fourier Transform

algorithms
- 7: Discrete Time Fourier Series (DTFS)

aliasing
- 10.1: Signal Sampling
- 10.5: Aliasing Phenomena

aliasing phenomena
- 10: Sampling and Reconstruction

analog
- 1.1: Signal Classifications and Properties

analog filters
- 11.9: Continuous Time Filter Design

analog signals
- 1.4: Common Continuous Time Signals

analysis
- 5.1: Introduction to Fourier Analysis
- 6.2: Continuous Time Fourier Series (CTFS)

AnalysisFormula
- 8.2: Continuous Time Fourier Transform (CTFT)

angles
- 15.4: Inner Products

angular frequency
- 1.8: Continuous Time Complex Exponential

aperiodic
- 1.1: Signal Classifications and Properties
- 8.1: Continuous Time Aperiodic Signals

aperiodic signals
- 8: Continuous Time Fourier Transform (CTFT)

AperiodicFunctions
- 7.1: Discrete Time Periodic Signals

aperiodicSignals
- 6.1: Continuous Time Periodic Signals
- 9: Discrete Time Fourier Transform (DTFT)
- 9.1: Discrete Time Aperiodic Signals

applications
- 13: Capstone Signal Processing Topics

ApplicationsOfDifferentialEquations
- 3.7: Linear Constant Coefficient Differential Equations

approximation
- 15: Appendix B- Hilbert Spaces Overview
- 15.14: Approximation and Projections in Hilbert Space

associativity
- 4.4: Properties of Discrete Time Convolution

asymptotes
- 11.7: Rational Functions and the Laplace Transform

audio
- 10.2: Sampling Theorem

B

bandlimited
- 10.5: Aliasing Phenomena

bandlimited signals
- 10.3: Signal Reconstruction
- 10.4: Perfect Reconstruction

basis
- 14: Appendix A- Linear Algebra Overview
- 14.1: Basic Linear Algebra
- 14.4: Eigen-Stuff in a Nutshell
- 15.12: Orthonormal Bases in Real and Complex Spaces

basis expansion
- 15.10: Function Space

basis functions
- 15.11: Haar Wavelet Basis

basis matrix
- 15.9: Orthonormal Basis Expansions

basis vectors
- 15.8: Types of Bases
- 15.10: Function Space

BIBO
- 4.6: BIBO Stability of Discrete Time Systems

BIBO stability
- 3.6: BIBO Stability of Continuous Time Systems

BIBO_stability
- 3: Time Domain Analysis of Continuous Time Systems

BIBOStability
- 4: Time Domain Analysis of Discrete Time Systems

Bilateral Laplace Transform
- 11.1: Laplace Transform

boundary
- 4.8: Solving Linear Constant Coefficient Difference Equations

boundary value problems
- 3.8: Solving Linear Constant Coefficient Differential Equations

boundaryconditions
- 3.7: Linear Constant Coefficient Differential Equations

bounded input
- 3.6: BIBO Stability of Continuous Time Systems
- 4.6: BIBO Stability of Discrete Time Systems

bounded output
- 3.6: BIBO Stability of Continuous Time Systems
- 4.6: BIBO Stability of Discrete Time Systems

Butterworth filters
- 11.9: Continuous Time Filter Design

C

calculation
- 14.4: Eigen-Stuff in a Nutshell

cascaded systems
- 2.2: Linear Time Invariant Systems

cauchy_schwarz_inequality
- 15: Appendix B- Hilbert Spaces Overview

causal
- 2.1: System Classifications and Properties

causal system
- 10.7: Discrete Time Processing of Continuous Time Signals

CDF
- 17: Appendix D- Viewing Interactive Content

change of basis
- 15.10: Function Space

characteristic polynomial
- 3.8: Solving Linear Constant Coefficient Differential Equations

Chebyshev filters
- 11.9: Continuous Time Filter Design

circular convolution
- 3.3: Continuous Time Convolution
- 4.3: Discrete Time Convolution
- 6.5: Continuous Time Circular Convolution and the CTFS
- 7: Discrete Time Fourier Series (DTFS)
- 7.5: Discrete Time Circular Convolution and the DTFS

classification
- 1: Introduction to Signals
- 2.1: System Classifications and Properties

classifications
- 2: Introduction to Systems

closest_point
- 15.14: Approximation and Projections in Hilbert Space

closure
- 15.2: Vector Spaces

coefficients
- 6.2: Continuous Time Fourier Series (CTFS)
- 8.2: Continuous Time Fourier Transform (CTFT)

collaboration
- InfoPage

CommonTransforms
- 9: Discrete Time Fourier Transform (DTFT)

communication
- 10.2: Sampling Theorem

communications applications
- 12.3: Properties of the Z-Transform

commutativity
- 4.4: Properties of Discrete Time Convolution

complete
- 15.5: Hilbert Spaces

complex
- 6.2: Continuous Time Fourier Series (CTFS)

complex analysis
- 11.4: Inverse Laplace Transform
- 12.4: Inverse Z-Transform

complex constants
- 3.8: Solving Linear Constant Coefficient Differential Equations

complex exponentials
1.4: Common Continuous Time Signals
1.5: Common Discrete Time Signals
1.8: Continuous Time Complex Exponential
1.9: Discrete Time Complex Exponential
3.5: Eigenfunctions of Continuous Time LTI Systems
6: Continuous Time Fourier Series (CTFS)
6.7: Gibbs Phenomena
7.2: Discrete Time Fourier Series (DTFS)
complex numbers
1.8: Continuous Time Complex Exponential
1.9: Discrete Time Complex Exponential
13.3: Deriving the Fast Fourier Transform
15.1: Fields and Complex Numbers
complex plane
12.9: Discrete Time Filter Design
15.1: Fields and Complex Numbers
complex sinusoids
6.6: Convergence of Fourier Series
7: Discrete Time Fourier Series (DTFS)
Complex Variables
11.1: Laplace Transform
complex vector space
15.2: Vector Spaces
complex_numbers
15: Appendix B- Hilbert Spaces Overview
ComplexExponentials
8.2: Continuous Time Fourier Transform (CTFT)
complexFunctions
15.5: Hilbert Spaces
complexity
13.1: DFT- Fast Fourier Transform
13.2: The Fast Fourier Transform (FFT)
13.3: Deriving the Fast Fourier Transform
ComplexScalars
15.7: Common Hilbert Spaces
complexValuedSignal
13.1: DFT- Fast Fourier Transform
compression
1.3: Signal Operations
computation
13.2: The Fast Fourier Transform (FFT)
computational efficiency
13.3: Deriving the Fast Fourier Transform
computational software
17.2: Getting Started with Mathematica
computationalSteps
13.1: DFT- Fast Fourier Transform
conjugate
15.12: Orthonormal Bases in Real and Complex Spaces
conjugateSymmetry
13.1: DFT- Fast Fourier Transform
conjugation
4.4: Properties of Discrete Time Convolution
Connexions
17: Appendix D- Viewing Interactive Content
continuity
16.1: Introduction to Metric Spaces
continuous
1: Introduction to Signals
2.1: System Classifications and Properties
6.2: Continuous Time Fourier Series (CTFS)
12.2: Common Z-Transforms
continuous functions
6.7: Gibbs Phenomena
continuous signals
8.5: Continuous Time Convolution and the CTFT
continuous time
1.4: Common Continuous Time Signals
1.6: Continuous Time Impulse Function
1.8: Continuous Time Complex Exponential
3.2: Continuous Time Impulse Response
3.6: BIBO Stability of Continuous Time Systems
6: Continuous Time Fourier Series (CTFS)
8: Continuous Time Fourier Transform (CTFT)
10.6: Anti-Aliasing Filters
continuous time convolution
3.3: Continuous Time Convolution
3.4: Properties of Continuous Time Convolution
Continuous Time Fourier Series
15.13: Plancharel and Parseval's Theorems
continuous time signal
10: Sampling and Reconstruction
10.3: Signal Reconstruction
continuous time signals
3.5: Eigenfunctions of Continuous Time LTI Systems
continuous time systems
3.1: Continuous Time Systems
continuous_time
8.4: Properties of the CTFT
continuous_time_fourier_series
6.4: Properties of the CTFS
continuous_time_fourier_transform
10.1: Signal Sampling
continuous_time_signals
10.1: Signal Sampling
continuous_time_systems
3: Time Domain Analysis of Continuous Time Systems
ContinuousTime
8.2: Continuous Time Fourier Transform (CTFT)
11: Laplace Transform and Continuous Time System Design
ContinuousTimeFourierSeries
6.1: Continuous Time Periodic Signals
ContinuousTimeFourierTransform
5.1: Introduction to Fourier Analysis
8.1: Continuous Time Aperiodic Signals
ContinuousTimeSignals
6.1: Continuous Time Periodic Signals
15.7: Common Hilbert Spaces
control theory
11.5: Poles and Zeros in the S-Plane
12.5: Poles and Zeros in the Z-Plane
convergence
6.6: Convergence of Fourier Series
11.2: Common Laplace Transforms
11.3: Properties of the Laplace Transform
12.2: Common Z-Transforms
12.6: Region of Convergence for the Z-Transform
16: Appendix C- Analysis Topics Overview
16.1: Introduction to Metric Spaces
16.2: Convergence of Sequences
16.3: Convergence of Sequences of Vectors
convergence properties
16.4: Uniform Convergence of Function Sequences
convolution
1.6: Continuous Time Impulse Function
3: Time Domain Analysis of Continuous Time Systems
3.2: Continuous Time Impulse Response
3.3: Continuous Time Convolution
3.4: Properties of Continuous Time Convolution
3.5: Eigenfunctions of Continuous Time LTI Systems
4: Time Domain Analysis of Discrete Time Systems
4.2: Discrete Time Impulse Response
4.3: Discrete Time Convolution
4.5: Eigenfunctions of Discrete Time LTI Systems
6.4: Properties of the CTFS
6.5: Continuous Time Circular Convolution and the CTFS
7.4: Properties of the DTFS
8: Continuous Time Fourier Transform (CTFT)
8.4: Properties of the CTFT
8.5: Continuous Time Convolution and the CTFT
9.4: Properties of the DTFT
9.5: Discrete Time Convolution and the DTFT
11.3: Properties of the Laplace Transform
12: Z-Transform and Discrete Time System Design
12.3: Properties of the Z-Transform
13: Capstone Signal Processing Topics
14.5: Eigenfunctions of LTI Systems
convolution integral
8.5: Continuous Time Convolution and the CTFT
convolution properties
4.4: Properties of Discrete Time Convolution
convolution theorem
8.5: Continuous Time Convolution and the CTFT
correlation
13.4: Matched Filter Detector
cosine
9.3: Common Discrete Time Fourier Transforms
11.2: Common Laplace Transforms
counter examples
16.3: Convergence of Sequences of Vectors
CTFS
6: Continuous Time Fourier Series (CTFS)
CTFT
8: Continuous Time Fourier Transform (CTFT)
8.3: Common Fourier Transforms
8.5: Continuous Time Convolution and the CTFT
curriculumdevelopment
InfoPage

D

data visualization
17.2: Getting Started with Mathematica
decomposition
15.8: Types of Bases
definition
15.2: Vector Spaces
16.2: Convergence of Sequences
delta
12.2: Common Z-Transforms
delta function
1.7: Discrete Time Impulse Function
8.3: Common Fourier Transforms
9.3: Common Discrete Time Fourier Transforms
12: Z-Transform and Discrete Time System Design
demo
17.1: Viewing Embedded LabVIEW Content in Connexions

determinants
14.2: Eigenvectors and Eigenvalues
deterministic
1.1: Signal Classifications and Properties
DFT
7.2: Discrete Time Fourier Series (DTFS)
7.3: Common Discrete Fourier Series
7.5: Discrete Time Circular Convolution and the DTFS
13: Capstone Signal Processing Topics
13.1: DFT- Fast Fourier Transform
13.2: The Fast Fourier Transform (FFT)
13.3: Deriving the Fast Fourier Transform
difference equation
4.2: Discrete Time Impulse Response
12.8: Difference Equations
difference equations
4.1: Discrete Time Systems
4.7: Linear Constant Coefficient Difference Equations
differential equation
3.2: Continuous Time Impulse Response
differential equations
3.1: Continuous Time Systems
differentialequations
3.7: Linear Constant Coefficient Differential Equations
11: Laplace Transform and Continuous Time System Design
11.8: Differential Equations
differentiation
3.4: Properties of Continuous Time Convolution
7.4: Properties of the DTFS
digital
1.1: Signal Classifications and Properties
10.2: Sampling Theorem
digital filter
12.9: Discrete Time Filter Design
17.1: Viewing Embedded LabVIEW Content in Connexions
digital filters
4.7: Linear Constant Coefficient Difference Equations
11.9: Continuous Time Filter Design
digital logic
10.7: Discrete Time Processing of Continuous Time Signals
digital signal processing
4.1: Discrete Time Systems
4.2: Discrete Time Impulse Response
10.4: Perfect Reconstruction
10.6: Anti-Aliasing Filters
12: Z-Transform and Discrete Time System Design
12.1: Z-Transform
13.2: The Fast Fourier Transform (FFT)
digital signals
13.4: Matched Filter Detector
digitalcomputers
10.1: Signal Sampling
dilation
1.3: Signal Operations
dimension
15.10: Function Space
Dirac delta function
1.4: Common Continuous Time Signals
1.6: Continuous Time Impulse Function
3.4: Properties of Continuous Time Convolution
Dirichlet conditions
6.7: Gibbs Phenomena
discontinuities
11.7: Rational Functions and the Laplace Transform
12.7: Rational Functions and the Z-Transform
discrete
2.1: System Classifications and Properties
4.8: Solving Linear Constant Coefficient Difference Equations
9.5: Discrete Time Convolution and the DTFT
discrete Fourier transform
13.3: Deriving the Fast Fourier Transform
discrete Fourier transforms
7: Discrete Time Fourier Series (DTFS)
7.3: Common Discrete Fourier Series
discrete time
1.5: Common Discrete Time Signals
1.7: Discrete Time Impulse Function
1.9: Discrete Time Complex Exponential
4.2: Discrete Time Impulse Response
4.6: BIBO Stability of Discrete Time Systems
4.7: Linear Constant Coefficient Difference Equations
10.6: Anti-Aliasing Filters
discrete time convolution
4.4: Properties of Discrete Time Convolution
discrete time fourier series
7: Discrete Time Fourier Series (DTFS)
Discrete Time Fourier Transform
9.2: Discrete Time Fourier Transform (DTFT)
discrete time periodic signals
7: Discrete Time Fourier Series (DTFS)
discrete time processing
10.4: Perfect Reconstruction
discrete time signal
10: Sampling and Reconstruction
10.3: Signal Reconstruction
discrete time signals
4.3: Discrete Time Convolution
4.5: Eigenfunctions of Discrete Time LTI Systems
discrete time systems
4.1: Discrete Time Systems
discrete_time_signals
10.1: Signal Sampling
DiscreteConvolution
9: Discrete Time Fourier Transform (DTFT)
DiscreteFourierTransform
5.1: Introduction to Fourier Analysis
DiscreteFrequency
7.1: Discrete Time Periodic Signals
discreteTime
7.1: Discrete Time Periodic Signals
9: Discrete Time Fourier Transform (DTFT)
9.1: Discrete Time Aperiodic Signals
DiscreteTimeConvolution
4: Time Domain Analysis of Discrete Time Systems
DiscreteTimeFourierSeries
7.1: Discrete Time Periodic Signals
DiscreteTimeFourierTransform
5.1: Introduction to Fourier Analysis
9.1: Discrete Time Aperiodic Signals
9.4: Properties of the DTFT
DiscreteTimeImpulseResponse
4: Time Domain Analysis of Discrete Time Systems
DiscreteTimeSignals
15.7: Common Hilbert Spaces
DiscreteTimeSystems
4: Time Domain Analysis of Discrete Time Systems
distance
16.1: Introduction to Metric Spaces
distributivity
4.4: Properties of Discrete Time Convolution
divergence
12.6: Region of Convergence for the Z-Transform
documentation
17.2: Getting Started with Mathematica
domain
11.7: Rational Functions and the Laplace Transform
12.7: Rational Functions and the Z-Transform
dotproduct
15.4: Inner Products
DSP
12.8: Difference Equations
DTFS
7: Discrete Time Fourier Series (DTFS)
7.2: Discrete Time Fourier Series (DTFS)
DTFT
9: Discrete Time Fourier Transform (DTFT)
9.2: Discrete Time Fourier Transform (DTFT)
9.3: Common Discrete Time Fourier Transforms
9.4: Properties of the DTFT
9.5: Discrete Time Convolution and the DTFT
12.9: Discrete Time Filter Design

E

educationaltechnology
InfoPage
eigenfunction
6: Continuous Time Fourier Series (CTFS)
eigenfunctions
1.9: Discrete Time Complex Exponential
3: Time Domain Analysis of Continuous Time Systems
14.5: Eigenfunctions of LTI Systems
eigenvalue
14.5: Eigenfunctions of LTI Systems
eigenvalues
3: Time Domain Analysis of Continuous Time Systems
3.1: Continuous Time Systems
3.5: Eigenfunctions of Continuous Time LTI Systems
4.5: Eigenfunctions of Discrete Time LTI Systems
14: Appendix A- Linear Algebra Overview
14.2: Eigenvectors and Eigenvalues
14.4: Eigen-Stuff in a Nutshell
eigenvector
14.4: Eigen-Stuff in a Nutshell
eigenvectors
3.5: Eigenfunctions of Continuous Time LTI Systems
4.5: Eigenfunctions of Discrete Time LTI Systems
14: Appendix A- Linear Algebra Overview
14.2: Eigenvectors and Eigenvalues
14.5: Eigenfunctions of LTI Systems
15.9: Orthonormal Basis Expansions

electrical engineering
TitlePage
3.3: Continuous Time Convolution
4.3: Discrete Time Convolution
4.7: Linear Constant Coefficient Difference Equations
15.1: Fields and Complex Numbers
embedded
17: Appendix D- Viewing Interactive Content
energy
1.2: Signal Size and Norms
7.4: Properties of the DTFS
energy_of_signal
6.4: Properties of the CTFS
engineering
1.6: Continuous Time Impulse Function
1.7: Discrete Time Impulse Function
8.1: Continuous Time Aperiodic Signals
11.4: Inverse Laplace Transform
11.5: Poles and Zeros in the S-Plane
epsilon
16.2: Convergence of Sequences
16.4: Uniform Convergence of Function Sequences
equations
4.8: Solving Linear Constant Coefficient Difference Equations
error energy
10.6: Anti-Aliasing Filters
Euclidean metric
16.1: Introduction to Metric Spaces
Euler's formula
1.8: Continuous Time Complex Exponential
1.9: Discrete Time Complex Exponential
even signals
6.3: Common Fourier Series
examples
15.2: Vector Spaces
16.3: Convergence of Sequences of Vectors
examples of convergence
16.4: Uniform Convergence of Function Sequences
exponential
8.3: Common Fourier Transforms
9.3: Common Discrete Time Fourier Transforms
11.2: Common Laplace Transforms
exponential functions
12: Z-Transform and Discrete Time System Design
exponentials
6.2: Continuous Time Fourier Series (CTFS)

F

fast algorithms
13.2: The Fast Fourier Transform (FFT)
Fast Fourier Transform
13: Capstone Signal Processing Topics
13.3: Deriving the Fast Fourier Transform
Fast_Fourier_Transform
13.4: Matched Filter Detector
feedback systems
12.1: Z-Transform
FFT
7.2: Discrete Time Fourier Series (DTFS)
13: Capstone Signal Processing Topics
13.2: The Fast Fourier Transform (FFT)
13.3: Deriving the Fast Fourier Transform
Fibonacci sequence
4.7: Linear Constant Coefficient Difference Equations
fields
15: Appendix B- Hilbert Spaces Overview
15.1: Fields and Complex Numbers
filter design
11.5: Poles and Zeros in the S-Plane
11.9: Continuous Time Filter Design
12.4: Inverse Z-Transform
12.5: Poles and Zeros in the Z-Plane
12.6: Region of Convergence for the Z-Transform
12.8: Difference Equations
12.9: Discrete Time Filter Design
FilterDesign
11: Laplace Transform and Continuous Time System Design
filtering
10.4: Perfect Reconstruction
filters
10.6: Anti-Aliasing Filters
12.1: Z-Transform
finite
1.2: Signal Size and Norms
finite dimensional
15.10: Function Space
FiniteEnergyFunction
15.7: Common Hilbert Spaces
FiniteLengthSignals
7.1: Discrete Time Periodic Signals
force
1.7: Discrete Time Impulse Function
Fourier
5.1: Introduction to Fourier Analysis
6.2: Continuous Time Fourier Series (CTFS)
7.4: Properties of the DTFS
9.5: Discrete Time Convolution and the DTFT
12.2: Common Z-Transforms
15.11: Haar Wavelet Basis
Fourier Analysis
5: Introduction to Fourier Analysis
11.1: Laplace Transform
Fourier coefficients
6: Continuous Time Fourier Series (CTFS)
6.3: Common Fourier Series
6.5: Continuous Time Circular Convolution and the CTFS
6.6: Convergence of Fourier Series
6.7: Gibbs Phenomena
7.2: Discrete Time Fourier Series (DTFS)
9.2: Discrete Time Fourier Transform (DTFT)
15.13: Plancharel and Parseval's Theorems
Fourier methods
3.2: Continuous Time Impulse Response
Fourier series
3.3: Continuous Time Convolution
4.5: Eigenfunctions of Discrete Time LTI Systems
6: Continuous Time Fourier Series (CTFS)
6.3: Common Fourier Series
6.5: Continuous Time Circular Convolution and the CTFS
6.6: Convergence of Fourier Series
6.7: Gibbs Phenomena
7: Discrete Time Fourier Series (DTFS)
7.2: Discrete Time Fourier Series (DTFS)
7.3: Common Discrete Fourier Series
7.5: Discrete Time Circular Convolution and the DTFS
9.2: Discrete Time Fourier Transform (DTFT)
14: Appendix A- Linear Algebra Overview
14.1: Basic Linear Algebra
14.5: Eigenfunctions of LTI Systems
15.8: Types of Bases
15.9: Orthonormal Basis Expansions
Fourier transform
3.5: Eigenfunctions of Continuous Time LTI Systems
4.5: Eigenfunctions of Discrete Time LTI Systems
8: Continuous Time Fourier Transform (CTFT)
8.3: Common Fourier Transforms
8.5: Continuous Time Convolution and the CTFT
10.5: Aliasing Phenomena
12.3: Properties of the Z-Transform
12.8: Difference Equations
14.5: Eigenfunctions of LTI Systems
Fourier Transforms
3.1: Continuous Time Systems
4.1: Discrete Time Systems
5: Introduction to Fourier Analysis
fourier_coefficients
6.4: Properties of the CTFS
Fourier_series
15.14: Approximation and Projections in Hilbert Space
fourier_transform
8.4: Properties of the CTFT
FourierAnalysis
8.1: Continuous Time Aperiodic Signals
9.1: Discrete Time Aperiodic Signals
15.7: Common Hilbert Spaces
FourierSeries
5.1: Introduction to Fourier Analysis
8.2: Continuous Time Fourier Transform (CTFT)
FourierTransform
8.2: Continuous Time Fourier Transform (CTFT)
9: Discrete Time Fourier Transform (DTFT)
FourierTransformProperties
9.4: Properties of the DTFT
FourierTransforms
15.5: Hilbert Spaces
frequency
1.9: Discrete Time Complex Exponential
9.5: Discrete Time Convolution and the DTFT
10.2: Sampling Theorem

frequency domain
4.3: Discrete Time Convolution
6.4: Properties of the CTFS
6.5: Continuous Time Circular Convolution and the CTFS
7.3: Common Discrete Fourier Series
7.5: Discrete Time Circular Convolution and the DTFS
8: Continuous Time Fourier Transform (CTFT)
8.3: Common Fourier Transforms
9.3: Common Discrete Time Fourier Transforms
frequency information
15.11: Haar Wavelet Basis
frequency response
11.9: Continuous Time Filter Design
12.5: Poles and Zeros in the Z-Plane
12.8: Difference Equations
12.9: Discrete Time Filter Design
frequency_domain
7.4: Properties of the DTFS
8.4: Properties of the CTFT
10.1: Signal Sampling
frequencyComponents
13.1: DFT- Fast Fourier Transform
frequencyDomain
6.1: Continuous Time Periodic Signals
9.1: Discrete Time Aperiodic Signals
frequencyresponse
11.8: Differential Equations
FrequencyShift
9.4: Properties of the DTFT
FrequencyShifting
11.3: Properties of the Laplace Transform
function
6.6: Convergence of Fourier Series
11.2: Common Laplace Transforms
16.2: Convergence of Sequences
function sequences
16: Appendix C- Analysis Topics Overview
16.4: Uniform Convergence of Function Sequences
function spaces
14.1: Basic Linear Algebra
function_space
15: Appendix B- Hilbert Spaces Overview
functionalAnalysis
15.5: Hilbert Spaces
functions
8.3: Common Fourier Transforms
fundamentalPeriod
6.1: Continuous Time Periodic Signals

G

geometric length
15.3: Norms
geometry
15.4: Inner Products
Gibb's phenomena
15.11: Haar Wavelet Basis
Gibb's phenomenon
6.3: Common Fourier Series
Gibbs phenomena
6: Continuous Time Fourier Series (CTFS)
Gibbs Phenomenon
6.7: Gibbs Phenomena
graphical understanding
11.6: Region of Convergence for the Laplace Transform
guide
17: Appendix D- Viewing Interactive Content

H

Haar wavelet
15.11: Haar Wavelet Basis
harmonic sinusoids
7.2: Discrete Time Fourier Series (DTFS)
9.2: Discrete Time Fourier Transform (DTFT)
Hermitian
15.12: Orthonormal Bases in Real and Complex Spaces
Hilbert Space
15.9: Orthonormal Basis Expansions
15.13: Plancharel and Parseval's Theorems
Hilbert spaces
15.8: Types of Bases
hilbert_spaces
15: Appendix B- Hilbert Spaces Overview
HilbertSpaces
15.5: Hilbert Spaces
15.7: Common Hilbert Spaces
homogeneous
4.8: Solving Linear Constant Coefficient Difference Equations
homogeneous solution
3.8: Solving Linear Constant Coefficient Differential Equations

I

ideal lowpass filter
10.4: Perfect Reconstruction
identity matrix
14.2: Eigenvectors and Eigenvalues
IIR filters
11.9: Continuous Time Filter Design
image compression
15.11: Haar Wavelet Basis
image processing
15.11: Haar Wavelet Basis
impulse convolution
3.4: Properties of Continuous Time Convolution
4.4: Properties of Discrete Time Convolution
impulse response
1.7: Discrete Time Impulse Function
3.1: Continuous Time Systems
3.2: Continuous Time Impulse Response
3.3: Continuous Time Convolution
3.5: Eigenfunctions of Continuous Time LTI Systems
3.6: BIBO Stability of Continuous Time Systems
4.1: Discrete Time Systems
4.2: Discrete Time Impulse Response
4.3: Discrete Time Convolution
4.5: Eigenfunctions of Discrete Time LTI Systems
4.6: BIBO Stability of Discrete Time Systems
8.5: Continuous Time Convolution and the CTFT
9.5: Discrete Time Convolution and the DTFT
10.7: Discrete Time Processing of Continuous Time Signals
14.5: Eigenfunctions of LTI Systems
impulse_response
3: Time Domain Analysis of Continuous Time Systems
ImpulseResponse
4: Time Domain Analysis of Discrete Time Systems
infinite
1.2: Signal Size and Norms
infinite dimensional
15.10: Function Space
infinite dimensional spaces
16.3: Convergence of Sequences of Vectors
initial conditions
12.8: Difference Equations
initial value problems
3.8: Solving Linear Constant Coefficient Differential Equations
InitialConditions
3.7: Linear Constant Coefficient Differential Equations
initialvalueproblems
11.8: Differential Equations
inner product
15.6: Cauchy-Schwarz Inequality
15.8: Types of Bases
15.12: Orthonormal Bases in Real and Complex Spaces
15.13: Plancharel and Parseval's Theorems
inner_products
15: Appendix B- Hilbert Spaces Overview
InnerProduct
15.4: Inner Products
15.7: Common Hilbert Spaces
innerProductSpace
15.5: Hilbert Spaces
input
3.2: Continuous Time Impulse Response
input signal
3.3: Continuous Time Convolution
4.3: Discrete Time Convolution
inputoutput
11.8: Differential Equations
installation
17.1: Viewing Embedded LabVIEW Content in Connexions
instructions
17.1: Viewing Embedded LabVIEW Content in Connexions
integer
9.3: Common Discrete Time Fourier Transforms
16.2: Convergence of Sequences
integer N
16.4: Uniform Convergence of Function Sequences
integral
6.6: Convergence of Fourier Series
integralEquations
15.5: Hilbert Spaces
integration
7.4: Properties of the DTFS
interactive content
17.2: Getting Started with Mathematica
intercepts
12.7: Rational Functions and the Z-Transform

interpolation
10: Sampling and Reconstruction
10.3: Signal Reconstruction
10.5: Aliasing Phenomena
interpretation
14.4: Eigen-Stuff in a Nutshell
introduction
2.1: System Classifications and Properties
9.2: Discrete Time Fourier Transform (DTFT)
16.1: Introduction to Metric Spaces
invariant
2: Introduction to Systems
inverse CTFT
8.5: Continuous Time Convolution and the CTFT
inverse DFT
7.5: Discrete Time Circular Convolution and the DTFS
Inverse Laplace Transform
11.1: Laplace Transform
11.4: Inverse Laplace Transform
InverseFT
8.2: Continuous Time Fourier Transform (CTFT)
InverseLaplaceTransform
11: Laplace Transform and Continuous Time System Design

K
Khan Academy
14.1: Basic Linear Algebra

L
l_1
1.2: Signal Size and Norms
l_2
1.2: Signal Size and Norms
l_infty
1.2: Signal Size and Norms
l_p
1.2: Signal Size and Norms
LabVIEW
17: Appendix D- Viewing Interactive Content
17.1: Viewing Embedded LabVIEW Content in Connexions
laplace
11.2: Common Laplace Transforms
Laplace transform
3.2: Continuous Time Impulse Response
3.8: Solving Linear Constant Coefficient Differential Equations
11.1: Laplace Transform
11.4: Inverse Laplace Transform
11.5: Poles and Zeros in the S-Plane
11.6: Region of Convergence for the Laplace Transform
11.7: Rational Functions and the Laplace Transform
laplacetransform
5.1: Introduction to Fourier Analysis
11: Laplace Transform and Continuous Time System Design
11.3: Properties of the Laplace Transform
11.8: Differential Equations
LibreTexts
InfoPage
limit
16.2: Convergence of Sequences
16.3: Convergence of Sequences of Vectors
Line
15.14: Approximation and Projections in Hilbert Space
linear
2: Introduction to Systems
2.1: System Classifications and Properties
2.2: Linear Time Invariant Systems
4.8: Solving Linear Constant Coefficient Difference Equations
linear algebra
3.1: Continuous Time Systems
3.5: Eigenfunctions of Continuous Time LTI Systems
4.1: Discrete Time Systems
4.5: Eigenfunctions of Discrete Time LTI Systems
14: Appendix A- Linear Algebra Overview
14.1: Basic Linear Algebra
14.2: Eigenvectors and Eigenvalues
14.4: Eigen-Stuff in a Nutshell
15.1: Fields and Complex Numbers
15.6: Cauchy-Schwarz Inequality
15.12: Orthonormal Bases in Real and Complex Spaces
linear combinations
15.9: Orthonormal Basis Expansions
linear constant coefficient difference equations
4.7: Linear Constant Coefficient Difference Equations
linear convolution
7.5: Discrete Time Circular Convolution and the DTFS
linear equations
3.8: Solving Linear Constant Coefficient Differential Equations
linear function space
15.2: Vector Spaces
linear independence
14: Appendix A- Linear Algebra Overview
14.2: Eigenvectors and Eigenvalues
15.9: Orthonormal Basis Expansions
linear operators
3.5: Eigenfunctions of Continuous Time LTI Systems
4.5: Eigenfunctions of Discrete Time LTI Systems
linear systems
2.2: Linear Time Invariant Systems
14.2: Eigenvectors and Eigenvalues
linear time invariant
1.6: Continuous Time Impulse Function
1.8: Continuous Time Complex Exponential
1.9: Discrete Time Complex Exponential
14.5: Eigenfunctions of LTI Systems
linear time invariant system
3.3: Continuous Time Convolution
linear time invariant systems
3.1: Continuous Time Systems
3.5: Eigenfunctions of Continuous Time LTI Systems
4.3: Discrete Time Convolution
14: Appendix A- Linear Algebra Overview
linear_constant_coefficient_differential_equations
3: Time Domain Analysis of Continuous Time Systems
linear_transformation
6.4: Properties of the CTFS
linearalgebra
15.4: Inner Products
LinearConstantCoefficientDifferenceEquations
4: Time Domain Analysis of Discrete Time Systems
LinearConstantCoefficientODEs
3.7: Linear Constant Coefficient Differential Equations
linearity
4.1: Discrete Time Systems
7.4: Properties of the DTFS
8.4: Properties of the CTFT
12.3: Properties of the Z-Transform
LinearSystem
11.3: Properties of the Laplace Transform
LinearTimeInvariant
4: Time Domain Analysis of Discrete Time Systems
11: Laplace Transform and Continuous Time System Design
Linux
17.2: Getting Started with Mathematica
lowpass
10.6: Anti-Aliasing Filters
lowpass filter
10.3: Signal Reconstruction
10.7: Discrete Time Processing of Continuous Time Signals
LTI
3.2: Continuous Time Impulse Response
4: Time Domain Analysis of Discrete Time Systems
4.2: Discrete Time Impulse Response
6.2: Continuous Time Fourier Series (CTFS)
9.5: Discrete Time Convolution and the DTFT
12.6: Region of Convergence for the Z-Transform
12.8: Difference Equations
LTI system
3.6: BIBO Stability of Continuous Time Systems
8.5: Continuous Time Convolution and the CTFT
11.6: Region of Convergence for the Laplace Transform
12.7: Rational Functions and the Z-Transform
14.4: Eigen-Stuff in a Nutshell
LTI systems
2.2: Linear Time Invariant Systems
4.1: Discrete Time Systems
6.6: Convergence of Fourier Series
7.2: Discrete Time Fourier Series (DTFS)
9.2: Discrete Time Fourier Transform (DTFT)
14: Appendix A- Linear Algebra Overview
14.5: Eigenfunctions of LTI Systems
15.9: Orthonormal Basis Expansions
15.11: Haar Wavelet Basis
LTI_Systems
3: Time Domain Analysis of Continuous Time Systems
8.2: Continuous Time Fourier Transform (CTFT)

M
magnitude response
11.5: Poles and Zeros in the S-Plane
12.5: Poles and Zeros in the Z-Plane
magnitudes
12.9: Discrete Time Filter Design

matched filter
13.4: Matched Filter Detector
15.6: Cauchy-Schwarz Inequality
matched filter detector
13: Capstone Signal Processing Topics
math
15.14: Approximation and Projections in Hilbert Space
Mathematica
17: Appendix D- Viewing Interactive Content
17.2: Getting Started with Mathematica
mathematical concept
15.3: Norms
mathematical formulation
6.6: Convergence of Fourier Series
mathematical functions
1.6: Continuous Time Impulse Function
16.4: Uniform Convergence of Function Sequences
mathematical justification
13: Capstone Signal Processing Topics
mathematical modeling
4.7: Linear Constant Coefficient Difference Equations
mathematical properties
15.10: Function Space
mathematical_theorems
6.4: Properties of the CTFS
mathematicalDefinition
9.1: Discrete Time Aperiodic Signals
MathematicalModeling
3.7: Linear Constant Coefficient Differential Equations
mathematics
5: Introduction to Fourier Analysis
6.5: Continuous Time Circular Convolution and the CTFS
11.2: Common Laplace Transforms
11.4: Inverse Laplace Transform
15.4: Inner Products
15.12: Orthonormal Bases in Real and Complex Spaces
matlab
11.1: Laplace Transform
matrices
14.2: Eigenvectors and Eigenvalues
14.5: Eigenfunctions of LTI Systems
matrix
14.4: Eigen-Stuff in a Nutshell
15.12: Orthonormal Bases in Real and Complex Spaces
matrix diagonalization
14: Appendix A- Linear Algebra Overview
14.3: Matrix Diagonalization
matrix operations
14.2: Eigenvectors and Eigenvalues
metric spaces
16: Appendix C- Analysis Topics Overview
16.1: Introduction to Metric Spaces
Microsoft Windows
17.1: Viewing Embedded LabVIEW Content in Connexions
modeling
TitlePage
modulation
7.4: Properties of the DTFS
8.4: Properties of the CTFT
9.4: Properties of the DTFT
mother wavelet
15.11: Haar Wavelet Basis
multilinearity
4.4: Properties of Discrete Time Convolution

N

nonlinear
2.1: System Classifications and Properties
nonuniform convergence
6.7: Gibbs Phenomena
norm
1: Introduction to Signals
15.5: Hilbert Spaces
16.3: Convergence of Sequences of Vectors
norm convergence
16: Appendix C- Analysis Topics Overview
normalized basis
15.8: Types of Bases
normed spaces
15.8: Types of Bases
normed vector space
16.3: Convergence of Sequences of Vectors
normedLinearSpace
15.5: Hilbert Spaces
NormedSpaces
15.7: Common Hilbert Spaces
norms
1.2: Signal Size and Norms
15: Appendix B- Hilbert Spaces Overview
15.3: Norms
16.1: Introduction to Metric Spaces
numericalComputation
13.2: The Fast Fourier Transform (FFT)
numericalmethods
13.1: DFT- Fast Fourier Transform
Nyquist
10.2: Sampling Theorem
10.5: Aliasing Phenomena
10.6: Anti-Aliasing Filters
Nyquist rate
10.4: Perfect Reconstruction
nyquist_shannon_sampling_theorem
10.1: Signal Sampling

O

odd signals
6.3: Common Fourier Series
OER
InfoPage
openeducationinitiatives
InfoPage
operations
1: Introduction to Signals
operator
15.12: Orthonormal Bases in Real and Complex Spaces
optimization
15.6: Cauchy-Schwarz Inequality
ordinary differential equations
3.8: Solving Linear Constant Coefficient Differential Equations
ordinarydifferentialequations
3.7: Linear Constant Coefficient Differential Equations
11.8: Differential Equations
orthogonal
15.14: Approximation and Projections in Hilbert Space
orthogonal basis
15.8: Types of Bases
orthogonality
15.4: Inner Products
orthonormal
15.12: Orthonormal Bases in Real and Complex Spaces
orthonormal basis
15.8: Types of Bases
15.9: Orthonormal Basis Expansions
15.13: Plancharel and Parseval's Theorems
orthonormal_basis_expansions
15: Appendix B- Hilbert Spaces Overview
output
3.2: Continuous Time Impulse Response
output computation
4.2: Discrete Time Impulse Response
output convergence
11.6: Region of Convergence for the Laplace Transform
output signal
3.3: Continuous Time Convolution
4.3: Discrete Time Convolution

P

parallel systems
2.2: Linear Time Invariant Systems
Parseval's Theorem
15.13: Plancharel and Parseval's Theorems
Parseval_relation
6.4: Properties of the CTFS
ParsevalsRelation
9.4: Properties of the DTFT
Partial Fraction Expansion
11.4: Inverse Laplace Transform
particular
4.8: Solving Linear Constant Coefficient Difference Equations
particular solution
3.8: Solving Linear Constant Coefficient Differential Equations
pattern detection
13.4: Matched Filter Detector
pattern matching
15.6: Cauchy-Schwarz Inequality
performanceAnalysis
13.2: The Fast Fourier Transform (FFT)
periodic
1.1: Signal Classifications and Properties
6.6: Convergence of Fourier Series
8.1: Continuous Time Aperiodic Signals

periodic functions
6.5: Continuous Time Circular Convolution and the CTFS
14.1: Basic Linear Algebra
periodic sequences
7.5: Discrete Time Circular Convolution and the DTFS
periodic signals
1.5: Common Discrete Time Signals
6: Continuous Time Fourier Series (CTFS)
6.5: Continuous Time Circular Convolution and the CTFS
9.2: Discrete Time Fourier Transform (DTFT)
periodic signals power
15.13: Plancharel and Parseval's Theorems
periodic_signals
15.14: Approximation and Projections in Hilbert Space
PeriodicFunction
6.1: Continuous Time Periodic Signals
7.1: Discrete Time Periodic Signals
periodicSignals
6.1: Continuous Time Periodic Signals
7.1: Discrete Time Periodic Signals
9.1: Discrete Time Aperiodic Signals
phasors
1.8: Continuous Time Complex Exponential
physics
3.7: Linear Constant Coefficient Differential Equations
Plancherel theorem
15.13: Plancharel and Parseval's Theorems
plugin
17.1: Viewing Embedded LabVIEW Content in Connexions
pointwise convergence
16: Appendix C- Analysis Topics Overview
16.4: Uniform Convergence of Function Sequences
pointwise multiplication
7.5: Discrete Time Circular Convolution and the DTFS
pole/zero plots
12.9: Discrete Time Filter Design
poles
4.6: BIBO Stability of Discrete Time Systems
11.5: Poles and Zeros in the S-Plane
11.6: Region of Convergence for the Laplace Transform
12.4: Inverse Z-Transform
12.6: Region of Convergence for the Z-Transform
poles and zeros
12.1: Z-Transform
PolesZeros
11: Laplace Transform and Continuous Time System Design
polos
12.5: Poles and Zeros in the Z-Plane
polynomial
11.7: Rational Functions and the Laplace Transform
polynomials
12.7: Rational Functions and the Z-Transform
15.10: Function Space
positive integers
16.2: Convergence of Sequences
power
1.2: Signal Size and Norms
problems
4.8: Solving Linear Constant Coefficient Difference Equations
processing
10.2: Sampling Theorem
programming
17.2: Getting Started with Mathematica
projection
15.4: Inner Products
15.14: Approximation and Projections in Hilbert Space
properties
1: Introduction to Signals
1.6: Continuous Time Impulse Function
1.7: Discrete Time Impulse Function
2: Introduction to Systems
2.1: System Classifications and Properties
3: Time Domain Analysis of Continuous Time Systems
3.4: Properties of Continuous Time Convolution
6: Continuous Time Fourier Series (CTFS)
8.3: Common Fourier Transforms
8.4: Properties of the CTFT
11.3: Properties of the Laplace Transform
12.3: Properties of the Z-Transform
12.6: Region of Convergence for the Z-Transform
15.2: Vector Spaces
properties of CTFT
8: Continuous Time Fourier Transform (CTFT)
properties of DTFS
7: Discrete Time Fourier Series (DTFS)
properties of norms
15.3: Norms
PropertiesOfDTFT
9: Discrete Time Fourier Transform (DTFT)

Q

QRCode
TitlePage
quadratic polynomials
15.10: Function Space
quantization
10.7: Discrete Time Processing of Continuous Time Signals
quantumMechanics
15.5: Hilbert Spaces
quotient
12.7: Rational Functions and the Z-Transform

R

radar
15.6: Cauchy-Schwarz Inequality
random
1.1: Signal Classifications and Properties
rational function
11.7: Rational Functions and the Laplace Transform
12.7: Rational Functions and the Z-Transform
RationalFunctions
11: Laplace Transform and Continuous Time System Design
real analysis
16.4: Uniform Convergence of Function Sequences
real numbers
15.1: Fields and Complex Numbers
15.3: Norms
16.2: Convergence of Sequences
real vector space
15.2: Vector Spaces
RealScalars
15.7: Common Hilbert Spaces
realValuedSignal
13.1: DFT- Fast Fourier Transform
Reconstruction
6.7: Gibbs Phenomena
10: Sampling and Reconstruction
10.5: Aliasing Phenomena
10.6: Anti-Aliasing Filters
reconstruction filters
10.3: Signal Reconstruction
reconstruction process
10.3: Signal Reconstruction
recurrence relation
4.7: Linear Constant Coefficient Difference Equations
recursive formula
12.8: Difference Equations
region
11.2: Common Laplace Transforms
12.2: Common Z-Transforms
region of convergence
3.6: BIBO Stability of Continuous Time Systems
11.6: Region of Convergence for the Laplace Transform
12.5: Poles and Zeros in the Z-Plane
regionofconvergence
11: Laplace Transform and Continuous Time System Design
12.6: Region of Convergence for the Z-Transform
regions of convergence
12: Z-Transform and Discrete Time System Design
Rice University
TitlePage
Richard Baraniuk
TitlePage
RLC circuit
3.1: Continuous Time Systems
ROC
11.6: Region of Convergence for the Laplace Transform
12.4: Inverse Z-Transform
roots
11.7: Rational Functions and the Laplace Transform
12.7: Rational Functions and the Z-Transform
Runtime Engine
17.1: Viewing Embedded LabVIEW Content in Connexions

S

sample reconstruction
10.7: Discrete Time Processing of Continuous Time Signals

sampling
10.1: Signal Sampling
10.2: Sampling Theorem
10.3: Signal Reconstruction
10.5: Aliasing Phenomena
10.6: Anti-Aliasing Filters
12.1: Z-Transform
sampling rate
10.4: Perfect Reconstruction
sampling theorem
10: Sampling and Reconstruction
sampling_period
10.1: Signal Sampling
sampling_rate
10.1: Signal Sampling
sawtooth wave
6.3: Common Fourier Series
sawtooth waveform
7.3: Common Discrete Fourier Series
scalar multiplication
15.1: Fields and Complex Numbers
scalars
15.2: Vector Spaces
scaling
1: Introduction to Signals
15.3: Norms
sequences
15.3: Norms
16: Appendix C- Analysis Topics Overview
16.2: Convergence of Sequences
16.4: Uniform Convergence of Function Sequences
series
6.2: Continuous Time Fourier Series (CTFS)
12.2: Common Z-Transforms
Shannon
10.2: Sampling Theorem
10.5: Aliasing Phenomena
shifting
1: Introduction to Signals
signal
1: Introduction to Signals
10.2: Sampling Theorem
11.2: Common Laplace Transforms
12.2: Common Z-Transforms
signal approximation
6.3: Common Fourier Series
6.6: Convergence of Fourier Series
7.3: Common Discrete Fourier Series
signal decomposition
1.4: Common Continuous Time Signals
1.5: Common Discrete Time Signals
15.9: Orthonormal Basis Expansions
signal denoising
8.5: Continuous Time Convolution and the CTFT
signal detection
15.6: Cauchy-Schwarz Inequality
signal energy
15.13: Plancharel and Parseval's Theorems
signal manipulation
1.3: Signal Operations
signal operations
1.3: Signal Operations
signal processing
TitlePage
2.2: Linear Time Invariant Systems
3.1: Continuous Time Systems
3.4: Properties of Continuous Time Convolution
4.3: Discrete Time Convolution
4.4: Properties of Discrete Time Convolution
5: Introduction to Fourier Analysis
6.4: Properties of the CTFS
6.5: Continuous Time Circular Convolution and the CTFS
6.7: Gibbs Phenomena
7.5: Discrete Time Circular Convolution and the DTFS
8: Continuous Time Fourier Transform (CTFT)
8.3: Common Fourier Transforms
10.3: Signal Reconstruction
10.5: Aliasing Phenomena
10.7: Discrete Time Processing of Continuous Time Signals
11.1: Laplace Transform
11.4: Inverse Laplace Transform
11.9: Continuous Time Filter Design
12.3: Properties of the Z-Transform
12.4: Inverse Z-Transform
13.3: Deriving the Fast Fourier Transform
13.4: Matched Filter Detector
15.1: Fields and Complex Numbers
15.6: Cauchy-Schwarz Inequality
16.1: Introduction to Metric Spaces
signal properties
1.5: Common Discrete Time Signals
signal reconstruction
10.3: Signal Reconstruction
10.4: Perfect Reconstruction
signal representation
1.4: Common Continuous Time Signals
1.5: Common Discrete Time Signals
signal sampling
10: Sampling and Reconstruction
signal_detection
13.4: Matched Filter Detector
signal_processing
8.4: Properties of the CTFT
SignalDemonstration
7.1: Discrete Time Periodic Signals
signalMapping
6.1: Continuous Time Periodic Signals
SignalPeriod
7.1: Discrete Time Periodic Signals
signalprocessing
7.2: Discrete Time Fourier Series (DTFS)
8.1: Continuous Time Aperiodic Signals
9: Discrete Time Fourier Transform (DTFT)
11: Laplace Transform and Continuous Time System Design
11.3: Properties of the Laplace Transform
11.8: Differential Equations
13.1: DFT- Fast Fourier Transform
13.2: The Fast Fourier Transform (FFT)
SignalProcessingLaboratory
9.4: Properties of the DTFT
signalRepresentation
9.1: Discrete Time Aperiodic Signals
signals
1.1: Signal Classifications and Properties
1.4: Common Continuous Time Signals
1.5: Common Discrete Time Signals
1.6: Continuous Time Impulse Function
1.7: Discrete Time Impulse Function
1.8: Continuous Time Complex Exponential
1.9: Discrete Time Complex Exponential
3.4: Properties of Continuous Time Convolution
5.1: Introduction to Fourier Analysis
6.4: Properties of the CTFS
7.4: Properties of the DTFS
8.1: Continuous Time Aperiodic Signals
8.2: Continuous Time Fourier Transform (CTFT)
9.1: Discrete Time Aperiodic Signals
9.5: Discrete Time Convolution and the DTFT
11.6: Region of Convergence for the Laplace Transform
12: Z-Transform and Discrete Time System Design
12.6: Region of Convergence for the Z-Transform
signals and systems
TitlePage
4.2: Discrete Time Impulse Response
4.7: Linear Constant Coefficient Difference Equations
SignalsAndSystems
3.7: Linear Constant Coefficient Differential Equations
signaltheory
8.1: Continuous Time Aperiodic Signals
SimpleHarmonicMotion
3.7: Linear Constant Coefficient Differential Equations
sinc function
8.3: Common Fourier Transforms
9.3: Common Discrete Time Fourier Transforms
10.4: Perfect Reconstruction
sine
11.2: Common Laplace Transforms
sinusoid
1.5: Common Discrete Time Signals
1.8: Continuous Time Complex Exponential
6.3: Common Fourier Series
sinusoid waveform
7.3: Common Discrete Fourier Series
SinusoidalSignal
6.1: Continuous Time Periodic Signals
7.1: Discrete Time Periodic Signals
sinusoids
1.4: Common Continuous Time Signals
size
1: Introduction to Signals
software
17: Appendix D- Viewing Interactive Content
solution
4.8: Solving Linear Constant Coefficient Difference Equations
solutionmethods
11.8: Differential Equations
solving_differential_equations
3: Time Domain Analysis of Continuous Time Systems
SolvingDifferenceEquations
4: Time Domain Analysis of Discrete Time Systems
span
14: Appendix A- Linear Algebra Overview

spectral lines
9.2: Discrete Time Fourier Transform (DTFT)
spectrum
10: Sampling and Reconstruction
10.5: Aliasing Phenomena
13.1: DFT- Fast Fourier Transform
square wave
6.3: Common Fourier Series
6.7: Gibbs Phenomena
7.3: Common Discrete Fourier Series
SquareSummable
15.7: Common Hilbert Spaces
stability
3.6: BIBO Stability of Continuous Time Systems
4.6: BIBO Stability of Discrete Time Systems
11.5: Poles and Zeros in the S-Plane
12.1: Z-Transform
12.5: Poles and Zeros in the Z-Plane
stable
2.1: System Classifications and Properties
subspace
14.1: Basic Linear Algebra
15.14: Approximation and Projections in Hilbert Space
superposition principle
2.2: Linear Time Invariant Systems
9.2: Discrete Time Fourier Transform (DTFT)
symmetry
12.3: Properties of the Z-Transform
15.4: Inner Products
SynthesisFormula
8.2: Continuous Time Fourier Transform (CTFT)
system
2: Introduction to Systems
system analysis
3.1: Continuous Time Systems
system properties
2.2: Linear Time Invariant Systems
system property
4.6: BIBO Stability of Discrete Time Systems
system response
3.8: Solving Linear Constant Coefficient Differential Equations
4.2: Discrete Time Impulse Response
systems
1.1: Signal Classifications and Properties
1.3: Signal Operations
1.6: Continuous Time Impulse Function
1.7: Discrete Time Impulse Function
1.8: Continuous Time Complex Exponential
1.9: Discrete Time Complex Exponential
2: Introduction to Systems
2.1: System Classifications and Properties
3.2: Continuous Time Impulse Response
3.4: Properties of Continuous Time Convolution
4.8: Solving Linear Constant Coefficient Difference Equations
5.1: Introduction to Fourier Analysis
6.2: Continuous Time Fourier Series (CTFS)
systemsanalysis
11.8: Differential Equations

T

target signal
13.4: Matched Filter Detector
template_matching
13.4: Matched Filter Detector
textbooks
InfoPage
theorem
10.2: Sampling Theorem
16.3: Convergence of Sequences of Vectors
Time
1: Introduction to Signals
2: Introduction to Systems
6.2: Continuous Time Fourier Series (CTFS)
9.5: Discrete Time Convolution and the DTFT
12.2: Common Z-Transforms
time domain
3.3: Continuous Time Convolution
6.5: Continuous Time Circular Convolution and the CTFS
7.3: Common Discrete Fourier Series
8: Continuous Time Fourier Transform (CTFT)
8.3: Common Fourier Transforms
9.3: Common Discrete Time Fourier Transforms
time invariance
2.2: Linear Time Invariant Systems
4.1: Discrete Time Systems
time invariant systems
4.5: Eigenfunctions of Discrete Time LTI Systems
time reversal
1.3: Signal Operations
time scaling
1.3: Signal Operations
time shift
3.4: Properties of Continuous Time Convolution
4.4: Properties of Discrete Time Convolution
time shifting
1.3: Signal Operations
time_scaling
8.4: Properties of the CTFT
time_shifting
8.4: Properties of the CTFT
TimeDifferentiation
9.4: Properties of the DTFT
11.3: Properties of the Laplace Transform
timeDomain
6.1: Continuous Time Periodic Signals
9.1: Discrete Time Aperiodic Signals
TimeScaling
11.3: Properties of the Laplace Transform
TimeShifting
9.4: Properties of the DTFT
11.3: Properties of the Laplace Transform
transfer function
3.6: BIBO Stability of Continuous Time Systems
4.6: BIBO Stability of Discrete Time Systems
11.5: Poles and Zeros in the S-Plane
11.7: Rational Functions and the Laplace Transform
11.9: Continuous Time Filter Design
12.1: Z-Transform
12.5: Poles and Zeros in the Z-Plane
12.8: Difference Equations
12.9: Discrete Time Filter Design
transfer functions
4.1: Discrete Time Systems
11.6: Region of Convergence for the Laplace Transform
transferfunction
11.8: Differential Equations
transform
7.4: Properties of the DTFS
9.5: Discrete Time Convolution and the DTFT
11.2: Common Laplace Transforms
transformation
14.4: Eigen-Stuff in a Nutshell
15.9: Orthonormal Basis Expansions
transforms
5.1: Introduction to Fourier Analysis
transpose
15.12: Orthonormal Bases in Real and Complex Spaces
triangle function
9.3: Common Discrete Time Fourier Transforms
triangle inequality
16.1: Introduction to Metric Spaces
Triangle Property
15.3: Norms
triangle wave
6.3: Common Fourier Series
triangle waveform
7.3: Common Discrete Fourier Series
trigonometric functions
1.9: Discrete Time Complex Exponential
12: Z-Transform and Discrete Time System Design
tutorial
17: Appendix D- Viewing Interactive Content
tutorials
17.2: Getting Started with Mathematica

U

uniform convergence
16: Appendix C- Analysis Topics Overview
16.3: Convergence of Sequences of Vectors
16.4: Uniform Convergence of Function Sequences
unilateral laplace transform
11.1: Laplace Transform
unit
12.2: Common Z-Transforms
unit impulse
1.5: Common Discrete Time Signals
1.6: Continuous Time Impulse Function
1.7: Discrete Time Impulse Function
unit impulses
1.4: Common Continuous Time Signals
unit step
1.4: Common Continuous Time Signals
1.5: Common Discrete Time Signals
9.3: Common Discrete Time Fourier Transforms
12: Z-Transform and Discrete Time System Design

V

vector
14.4: Eigen-Stuff in a Nutshell
15.3: Norms
15.14: Approximation and Projections in Hilbert Space
16.2: Convergence of Sequences
vector addition
15.1: Fields and Complex Numbers

vector space
1.2: Signal Size and Norms
15.2: Vector Spaces
15.3: Norms
15.6: Cauchy-Schwarz Inequality
16.1: Introduction to Metric Spaces
vector spaces
14.2: Eigenvectors and Eigenvalues
15.1: Fields and Complex Numbers
15.8: Types of Bases
15.10: Function Space
vector_spaces
15: Appendix B- Hilbert Spaces Overview
vectors
15.2: Vector Spaces
15.4: Inner Products
16: Appendix C- Analysis Topics Overview
16.3: Convergence of Sequences of Vectors
VI
17.1: Viewing Embedded LabVIEW Content in Connexions
virtual instrument
17.1: Viewing Embedded LabVIEW Content in Connexions
W
waveform matching
13.4: Matched Filter Detector
waveforms
6: Continuous Time Fourier Series (CTFS)
wavelets
15.11: Haar Wavelet Basis
windowing
10.7: Discrete Time Processing of Continuous Time Signals
Windows
17.2: Getting Started with Mathematica
Wolfram Research
17.2: Getting Started with Mathematica
Z
zeros
11.5: Poles and Zeros in the S-Plane
12.4: Inverse Z-Transform
12.5: Poles and Zeros in the Z-Plane
zplane
12.6: Region of Convergence for the Z-Transform
ztransform
12.6: Region of Convergence for the Z-Transform

Glossary

Sample Word 1 | Sample Definition 1

Detailed Licensing

Overview

Title: Signals and Systems (Baraniuk et al.)

Webpages: 132

All licenses found:

- CC BY 3.0: 94.7% (125 pages)
- Undeclared: 5.3% (7 pages)

By Page

- Signals and Systems (Baraniuk et al.) - *CC BY 3.0*
  - Front Matter - *CC BY 3.0*
    - TitlePage - *Undeclared*
    - InfoPage - *CC BY 3.0*
    - Table of Contents - *Undeclared*
    - Licensing - *Undeclared*
    - About the Book - *Undeclared*
  - 1: Introduction to Signals - *CC BY 3.0*
    - 1.1: Signal Classifications and Properties - *CC BY 3.0*
    - 1.2: Signal Size and Norms - *CC BY 3.0*
    - 1.3: Signal Operations - *CC BY 3.0*
    - 1.4: Common Continuous Time Signals - *CC BY 3.0*
    - 1.5: Common Discrete Time Signals - *CC BY 3.0*
    - 1.6: Continuous Time Impulse Function - *CC BY 3.0*
    - 1.7: Discrete Time Impulse Function - *CC BY 3.0*
    - 1.8: Continuous Time Complex Exponential - *CC BY 3.0*
    - 1.9: Discrete Time Complex Exponential - *CC BY 3.0*
  - 2: Introduction to Systems - *CC BY 3.0*
    - 2.1: System Classifications and Properties - *CC BY 3.0*
    - 2.2: Linear Time Invariant Systems - *CC BY 3.0*
  - 3: Time Domain Analysis of Continuous Time Systems - *CC BY 3.0*
    - 3.1: Continuous Time Systems - *CC BY 3.0*
    - 3.2: Continuous Time Impulse Response - *CC BY 3.0*
    - 3.3: Continuous Time Convolution - *CC BY 3.0*
    - 3.4: Properties of Continuous Time Convolution - *CC BY 3.0*
    - 3.5: Eigenfunctions of Continuous Time LTI Systems - *CC BY 3.0*
    - 3.6: BIBO Stability of Continuous Time Systems - *CC BY 3.0*
    - 3.7: Linear Constant Coefficient Differential Equations - *CC BY 3.0*
    - 3.8: Solving Linear Constant Coefficient Differential Equations - *CC BY 3.0*
  - 4: Time Domain Analysis of Discrete Time Systems - *CC BY 3.0*
    - 4.1: Discrete Time Systems - *CC BY 3.0*
    - 4.2: Discrete Time Impulse Response - *CC BY 3.0*
    - 4.3: Discrete Time Convolution - *CC BY 3.0*
    - 4.4: Properties of Discrete Time Convolution - *CC BY 3.0*
    - 4.5: Eigenfunctions of Discrete Time LTI Systems - *CC BY 3.0*
    - 4.6: BIBO Stability of Discrete Time Systems - *CC BY 3.0*
    - 4.7: Linear Constant Coefficient Difference Equations - *CC BY 3.0*
    - 4.8: Solving Linear Constant Coefficient Difference Equations - *CC BY 3.0*
  - 5: Introduction to Fourier Analysis - *CC BY 3.0*
    - 5.1: Introduction to Fourier Analysis - *CC BY 3.0*
  - 6: Continuous Time Fourier Series (CTFS) - *CC BY 3.0*
    - 6.1: Continuous Time Periodic Signals - *CC BY 3.0*
    - 6.2: Continuous Time Fourier Series (CTFS) - *CC BY 3.0*
    - 6.3: Common Fourier Series - *CC BY 3.0*
    - 6.4: Properties of the CTFS - *CC BY 3.0*
    - 6.5: Continuous Time Circular Convolution and the CTFS - *CC BY 3.0*
    - 6.6: Convergence of Fourier Series - *CC BY 3.0*
    - 6.7: Gibbs Phenomena - *CC BY 3.0*
  - 7: Discrete Time Fourier Series (DTFS) - *CC BY 3.0*
    - 7.1: Discrete Time Periodic Signals - *CC BY 3.0*
    - 7.2: Discrete Time Fourier Series (DTFS) - *CC BY 3.0*
    - 7.3: Common Discrete Fourier Series - *CC BY 3.0*
    - 7.4: Properties of the DTFS - *CC BY 3.0*

- 7.5: Discrete Time Circular Convolution and the DTFS - *CC BY 3.0*

- 8: Continuous Time Fourier Transform (CTFT) - *CC BY 3.0*
  - 8.1: Continuous Time Aperiodic Signals - *CC BY 3.0*
  - 8.2: Continuous Time Fourier Transform (CTFT) - *CC BY 3.0*
  - 8.3: Common Fourier Transforms - *CC BY 3.0*
  - 8.4: Properties of the CTFT - *CC BY 3.0*
  - 8.5: Continuous Time Convolution and the CTFT - *CC BY 3.0*
- 9: Discrete Time Fourier Transform (DTFT) - *CC BY 3.0*
  - 9.1: Discrete Time Aperiodic Signals - *CC BY 3.0*
  - 9.2: Discrete Time Fourier Transform (DTFT) - *CC BY 3.0*
  - 9.3: Common Discrete Time Fourier Transforms - *CC BY 3.0*
  - 9.4: Properties of the DTFT - *CC BY 3.0*
  - 9.5: Discrete Time Convolution and the DTFT - *CC BY 3.0*
- 10: Sampling and Reconstruction - *CC BY 3.0*
  - 10.1: Signal Sampling - *CC BY 3.0*
  - 10.2: Sampling Theorem - *CC BY 3.0*
  - 10.3: Signal Reconstruction - *CC BY 3.0*
  - 10.4: Perfect Reconstruction - *CC BY 3.0*
  - 10.5: Aliasing Phenomena - *CC BY 3.0*
  - 10.6: Anti-Aliasing Filters - *CC BY 3.0*
  - 10.7: Discrete Time Processing of Continuous Time Signals - *CC BY 3.0*
- 11: Laplace Transform and Continuous Time System Design - *CC BY 3.0*
  - 11.1: Laplace Transform - *CC BY 3.0*
  - 11.2: Common Laplace Transforms - *CC BY 3.0*
  - 11.3: Properties of the Laplace Transform - *CC BY 3.0*
  - 11.4: Inverse Laplace Transform - *CC BY 3.0*
  - 11.5: Poles and Zeros in the S-Plane - *CC BY 3.0*
  - 11.6: Region of Convergence for the Laplace Transform - *CC BY 3.0*
  - 11.7: Rational Functions and the Laplace Transform - *CC BY 3.0*
  - 11.8: Differential Equations - *CC BY 3.0*
  - 11.9: Continuous Time Filter Design - *CC BY 3.0*
- 12: Z-Transform and Discrete Time System Design - *CC BY 3.0*
  - 12.1: Z-Transform - *CC BY 3.0*
  - 12.2: Common Z-Transforms - *CC BY 3.0*
  - 12.3: Properties of the Z-Transform - *CC BY 3.0*
  - 12.4: Inverse Z-Transform - *CC BY 3.0*
  - 12.5: Poles and Zeros in the Z-Plane - *CC BY 3.0*
  - 12.6: Region of Convergence for the Z-Transform - *CC BY 3.0*
  - 12.7: Rational Functions and the Z-Transform - *CC BY 3.0*
  - 12.8: Difference Equations - *CC BY 3.0*
  - 12.9: Discrete Time Filter Design - *CC BY 3.0*
- 13: Capstone Signal Processing Topics - *CC BY 3.0*
  - 13.1: DFT- Fast Fourier Transform - *CC BY 3.0*
  - 13.2: The Fast Fourier Transform (FFT) - *CC BY 3.0*
  - 13.3: Deriving the Fast Fourier Transform - *CC BY 3.0*
  - 13.4: Matched Filter Detector - *CC BY 3.0*
- 14: Appendix A- Linear Algebra Overview - *CC BY 3.0*
  - 14.1: Basic Linear Algebra - *CC BY 3.0*
  - 14.2: Eigenvectors and Eigenvalues - *CC BY 3.0*
  - 14.3: Matrix Diagonalization - *CC BY 3.0*
  - 14.4: Eigen-Stuff in a Nutshell - *CC BY 3.0*
  - 14.5: Eigenfunctions of LTI Systems - *CC BY 3.0*
- 15: Appendix B- Hilbert Spaces Overview - *CC BY 3.0*
  - 15.1: Fields and Complex Numbers - *CC BY 3.0*
  - 15.2: Vector Spaces - *CC BY 3.0*
  - 15.3: Norms - *CC BY 3.0*
  - 15.4: Inner Products - *CC BY 3.0*
  - 15.5: Hilbert Spaces - *CC BY 3.0*
  - 15.6: Cauchy-Schwarz Inequality - *CC BY 3.0*
  - 15.7: Common Hilbert Spaces - *CC BY 3.0*
  - 15.8: Types of Bases - *CC BY 3.0*
  - 15.9: Orthonormal Basis Expansions - *CC BY 3.0*
  - 15.10: Function Space - *CC BY 3.0*
  - 15.11: Haar Wavelet Basis - *CC BY 3.0*
  - 15.12: Orthonormal Bases in Real and Complex Spaces - *CC BY 3.0*
  - 15.13: Plancharel and Parseval's Theorems - *CC BY 3.0*
  - 15.14: Approximation and Projections in Hilbert Space - *CC BY 3.0*
- 16: Appendix C- Analysis Topics Overview - *CC BY 3.0*
  - 16.1: Introduction to Metric Spaces - *CC BY 3.0*
  - 16.2: Convergence of Sequences - *CC BY 3.0*
  - 16.3: Convergence of Sequences of Vectors - *CC BY 3.0*
  - 16.4: Uniform Convergence of Function Sequences - *CC BY 3.0*
- 17: Appendix D- Viewing Interactive Content - *CC BY 3.0*

- 17.1: Viewing Embedded LabVIEW Content in Connexions - *CC BY 3.0*
- 17.2: Getting Started with Mathematica - *CC BY 3.0*

Back Matter - *CC BY 3.0*

- Index - *Undeclared*
- Glossary - *Undeclared*
- Detailed Licensing - *Undeclared*