Lecture Slides for Signals and Systems

Edition 4.0

Michael D. Adams

Department of Electrical and Computer Engineering
University of Victoria
Victoria, British Columbia, Canada

To obtain the most recent version of these lecture slides (with functional hyperlinks) or for additional information and resources related to these slides (including video lectures and errata), please visit:

http://www.ece.uvic.ca/~mdadams/sigsysbook

If you like these lecture slides, **please consider posting a review** of them at:

https://play.google.com/store/search?q=ISBN:9780987919793 or
http://books.google.com/books?vid=ISBN9780987919793

youtube.com/iamcanadian1867 github.com/mdadams @mdadams16

This document was typeset with LaTeX.

ISBN 978-0-9879197-9-3 (PDF)

License I

License II

in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this

License III

License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections; and,

b. to Distribute and Publicly Perform the Work including as incorporated in Collections.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Adaptations. Subject to 8(f), all rights not expressly granted by Licensor

are hereby reserved, including but not limited to the rights set forth in Section 4(d).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested.
b. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.
c. If You Distribute, or Publicly Perform the Work or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice,

terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Collection, at a minimum such credit will appear, if a credit for all contributing authors of Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

d. For the avoidance of doubt:

   i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
   ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License if Your exercise of such rights is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(b) and otherwise waives the right to collect royalties through any statutory or compulsory licensing scheme; and,
   iii. Voluntary License Schemes. The Licensor reserves the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License that is for a purpose or use which is otherwise than noncommercial as permitted

License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
e. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

License VIII

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at http://creativecommons.org/.

Other Textbooks and Lecture Slides by the Author I

1. M. D. Adams, *Exercises for Programming in C++ (Version 2021-04-01)*, Apr. 2021, xxii + 136 pages, ISBN 978-0-9879197-5-5 (PDF). Available from Google Books, Google Play Books, and author's web site http://www.ece.uvic.ca/~mdadams/cppbook.
2. M. D. Adams, *Lecture Slides for Programming in C++ (Version 2021-04-01)*, Apr. 2021, xxiii + 2901 slides, ISBN 978-0-9879197-4-8 (PDF). Available from Google Books, Google Play Books, and author's web site http://www.ece.uvic.ca/~mdadams/cppbook.
3. M. D. Adams, *Multiresolution Signal and Geometry Processing: Filter Banks, Wavelets, and Subdivision (Version 2013-09-26)*, University of Victoria, Victoria, BC, Canada, Sept. 2013, xxxviii + 538 pages, ISBN 978-1-55058-507-0 (print), ISBN 978-1-55058-508-7 (PDF). Available from Google Books, Google Play Books, University of Victoria Bookstore, and author's web site http://www.ece.uvic.ca/~mdadams/waveletbook.

Other Textbooks and Lecture Slides by the Author II

4 M. D. Adams, *Lecture Slides for Multiresolution Signal and Geometry Processing (Version 2015-02-03)*, University of Victoria, Victoria, BC, Canada, Feb. 2015, xi + 587 slides, ISBN 978-1-55058-535-3 (print), ISBN 978-1-55058-536-0 (PDF). Available from Google Books, Google Play Books, University of Victoria Bookstore, and author's web site http://www.ece.uvic.ca/~mdadams/waveletbook.

5 M. D. Adams, *Signals and Systems*, Edition 4.0, Jan. 2022, xliv + 690 pages, ISBN 978-0-9879197-7-9 (PDF). Available from Google Books, Google Play Books, University of Victoria Bookstore, and author's web site http://www.ece.uvic.ca/~mdadams/sigsysbook.

Part 0

Preface

About These Lecture Slides

- This document constitutes a detailed set of lecture slides on signals and systems, covering both the continuous-time and discrete-time cases.
- These slides are organized in such a way as to facilitate the teaching of a course that covers:
  - only the continuous-time case, or
  - only the discrete-time case, or
  - both the continuous-time and discrete-time cases.
- These slides are intended to be used in conjunction with the following textbook:
  - M. D. Adams, *Signals and Systems*, Edition 4.0, Jan. 2022, xliv + 690 pages, ISBN 978-0-9879197-7-9 (PDF). Available from Google Books, Google Play Books, University of Victoria Bookstore, and author's web site http://www.ece.uvic.ca/~mdadams/sigsysbook.

Typesetting Conventions

- In a definition, the term being defined is often typeset in a font **like this**.
- To emphasize particular words, the words are typeset in a font ***like this***.
- To show that particular text is associated with a hyperlink to an internal target, the text is typeset like this.
- To show that particular text is associated with a hyperlink to an external document, the text is typeset like this.
- URLs are typeset like `http://www.ece.uvic.ca/~mdadams`.

Companion Web Site

- The author of the lecture slides maintains a companion web site for the lecture slides and the associated textbook.
- The most recent version of the textbook and lecture slides can be downloaded from this site.
- Additional information related to the slides is also available from this site, including:
  - errata for the slides; and
  - information on the companion web site, companion Git repository, and companion YouTube channel for the slides.
- The URL of this web site is:
  - `http://www.ece.uvic.ca/~mdadams/sigsysbook`

Video Lectures

- The author has prepared video lectures for some of the material covered in these slides and the associated textbook.
- All of the videos are hosted by YouTube and available through the author's YouTube channel:
  - https://www.youtube.com/iamcanadian1867
- The most up-to-date information about this video-lecture content can be found at:
  - https://www.ece.uvic.ca/~mdadams/sigsysbook/#video_lectures

Companion Git Repository

- These lecture slides have a companion Git repository.
- Numerous code examples are available from this repository.
- This repository is hosted by GitHub.
- The URL of the main repository page on GitHub is:
  - https://github.com/mdadams/sigsysbook_companion
- The URL of the actual repository itself is:
  - https://github.com/mdadams/sigsysbook_companion.git

Part 1

Introduction

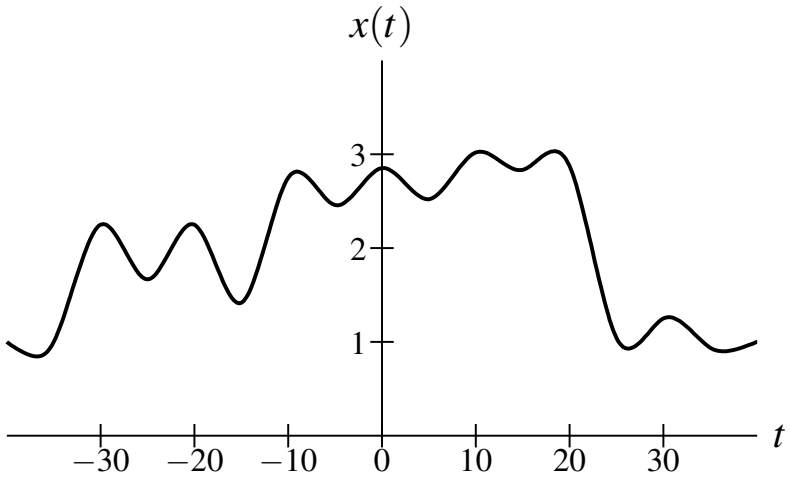
Signals

- A **signal** is a function of one or more variables that conveys information about some (usually physical) phenomenon.
- For a function f, in the expression $f(t_1, t_2, \ldots, t_n)$, each of the $\{t_k\}$ is called an **independent variable**, while the function value itself is referred to as a **dependent variable**.
- Some examples of signals include:
  - a voltage or current in an electronic circuit
  - the position, velocity, or acceleration of an object
  - a force or torque in a mechanical system
  - a flow rate of a liquid or gas in a chemical process
  - a digital image, digital video, or digital audio
  - a stock market index

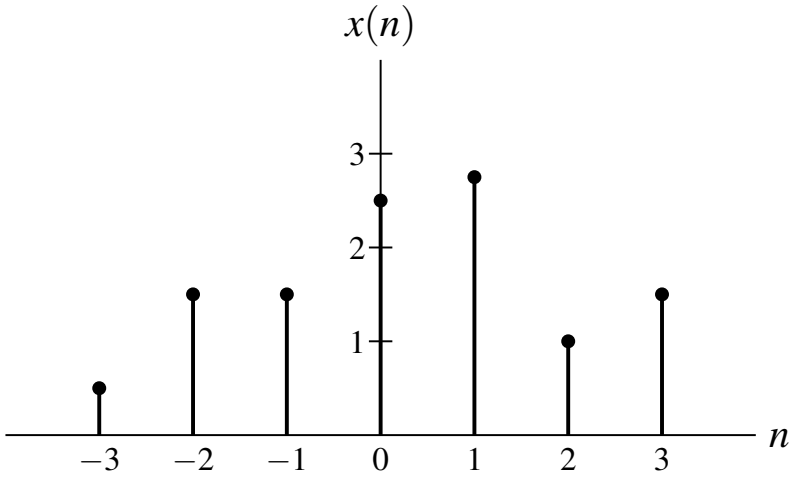
Classification of Signals

- Number of independent variables (i.e., dimensionality):
  - A signal with ***one*** independent variable is said to be **one dimensional** (e.g., audio).
  - A signal with ***more than one*** independent variable is said to be **multi-dimensional** (e.g., image).
- Continuous or discrete independent variables:
  - A signal with ***continuous*** independent variables is said to be **continuous time (CT)** (e.g., voltage waveform).
  - A signal with ***discrete*** independent variables is said to be **discrete time (DT)** (e.g., stock market index).
- Continuous or discrete dependent variable:
  - A signal with a ***continuous*** dependent variable is said to be **continuous valued** (e.g., voltage waveform).
  - A signal with a ***discrete*** dependent variable is said to be **discrete valued** (e.g., digital image).
- A ***continuous-valued CT*** signal is said to be **analog** (e.g., voltage waveform).
- A ***discrete-valued DT*** signal is said to be **digital** (e.g., digital audio).

Graphical Representation of Signals



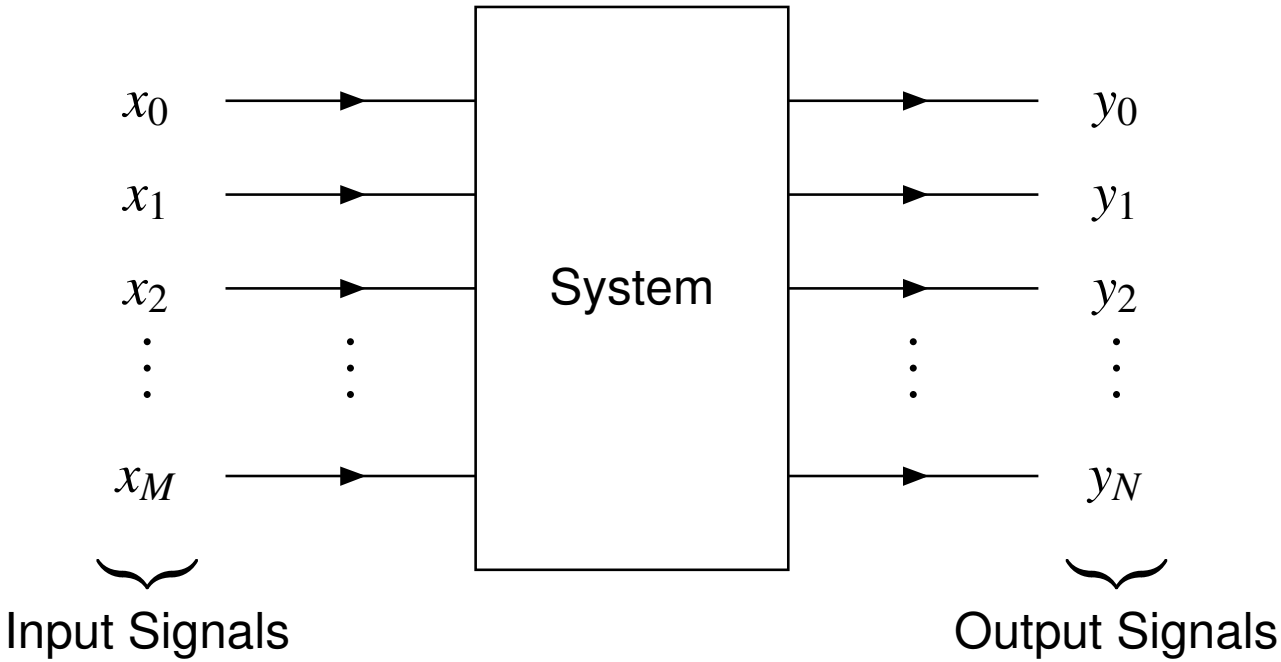
Continuous-Time (CT) Signal



Discrete-Time (DT) Signal

Systems

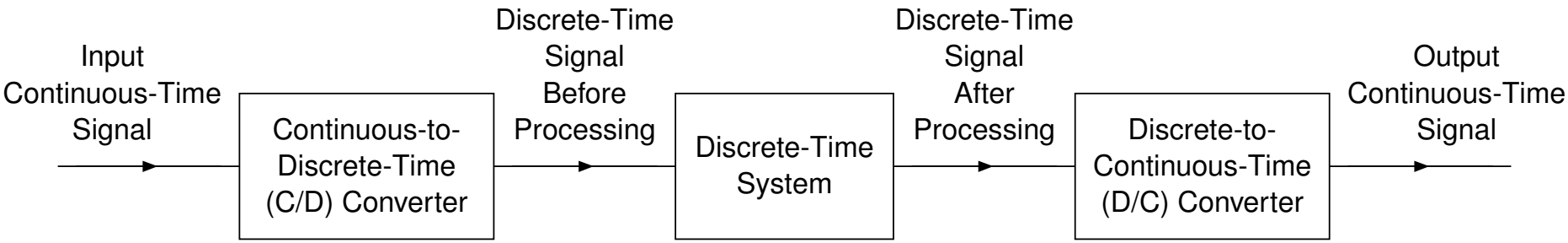
- A **system** is an entity that processes one or more input signals in order to produce one or more output signals.



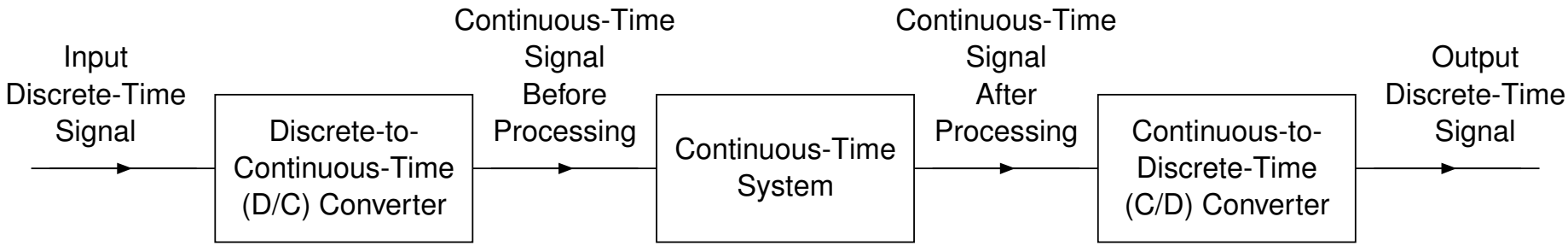
Classification of Systems

- Number of inputs:
  - A system with ***one*** input is said to be **single input (SI)**.
  - A system with ***more than one*** input is said to be **multiple input (MI)**.
- Number of outputs:
  - A system with ***one*** output is said to be **single output (SO)**.
  - A system with ***more than one*** output is said to be **multiple output (MO)**.
- Types of signals processed:
  - A system can be classified in terms of the ***types of signals*** that it processes.
  - Consequently, terms such as the following (which describe signals) can also be used to describe systems:
    - one-dimensional and multi-dimensional,
    - continuous-time (CT) and discrete-time (DT), and
    - analog and digital.
  - For example, a continuous-time (CT) system processes CT signals and a discrete-time (DT) system processes DT signals.

Signal Processing Systems

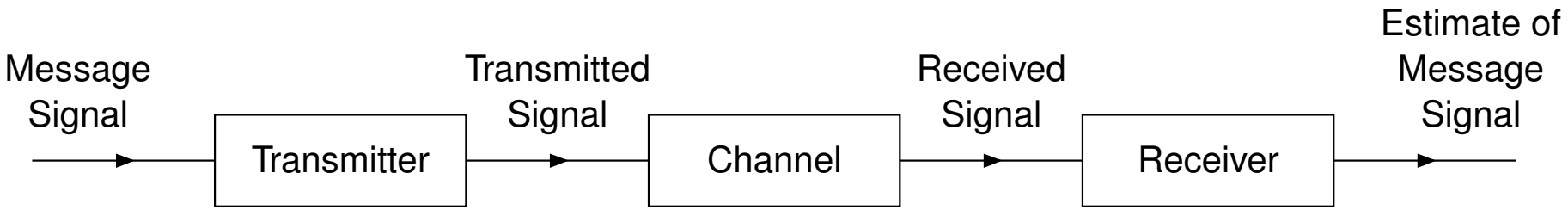


Processing a Continuous-Time Signal With a Discrete-Time System



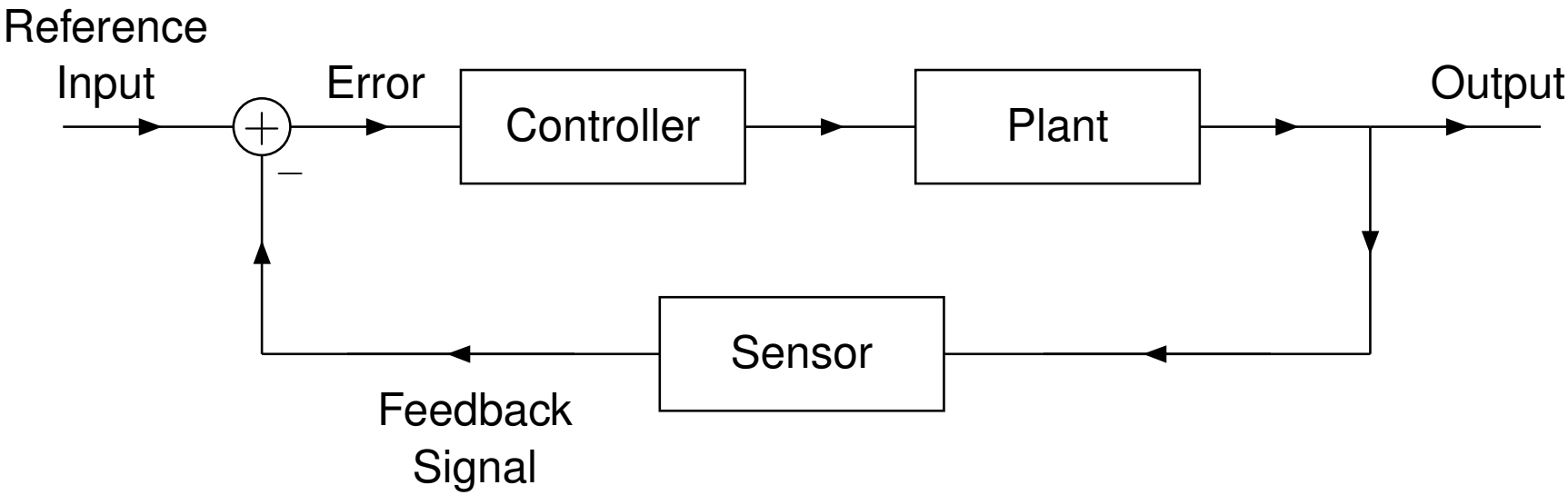
Processing a Discrete-Time Signal With a Continuous-Time System

Communication Systems



General Structure of a Communication System

Control Systems



General Structure of a Feedback Control System

Why Study Signals and Systems?

- Engineers build systems that process/manipulate signals.
- We need a formal mathematical framework for the study of such systems.
- Such a framework is necessary in order to ensure that a system will meet the required specifications (e.g., performance and safety).
- If a system fails to meet the required specifications or fails to work altogether, negative consequences usually ensue.
- When a system fails to operate as expected, the consequences can sometimes be catastrophic.

System Failure Example: Tacoma Narrows Bridge

- The (original) Tacoma Narrows Bridge was a suspension bridge linking Tacoma and Gig Harbor (WA, USA).
- This mile-long bridge, with a 2,800-foot main span, was the third largest suspension bridge at the time of opening.
- Construction began in Nov. 1938 and took about 19 months to build at a cost of $6,400,000.
- On July 1, 1940, the bridge opened to traffic.
- On Nov. 7, 1940 at approximately 11:00, the bridge collapsed during a moderate (42 miles/hour) wind storm.
- The bridge was supposed to withstand winds of up to 120 miles/hour.
- The collapse was due to wind-induced vibrations and an ***unstable mechanical system***.
- Repair of the bridge was not possible.
- Fortunately, a dog trapped in an abandoned car was the only fatality.

System Failure Example: Tacoma Narrows Bridge (Continued)

Image of bridge collapse omitted for copyright reasons.

A video of the bridge collapse can be found at
https://youtu.be/j-zczJXSxnw.

Part 2

Preliminaries

Section 2.1

Functions, Sequences, System Operators, and Transforms

- A **rational number** is a number of the form x/y, where x and y are integers and $y \neq 0$ (i.e., a ratio of integers).
- For example, $-\frac{5}{3}$, $\frac{17}{11}$, and $0 = \frac{0}{1}$ are rational numbers, whereas π and e are irrational numbers (i.e., not rational).
- The symbols employed to denote several commonly-used sets are as follows:

Symbol	Set
$\mathbb{Z}$	integers
$\mathbb{R}$	real numbers
$\mathbb{C}$	complex numbers
$\mathbb{Q}$	rational numbers

Notation for Sets of Consecutive Integers

- For two integers a and b, we define the following notation for sets of consecutive integers:

$$\begin{aligned}
[a\,..\,b] &= \{x \in \mathbb{Z} : a \leq x \leq b\}, \\
[a\,..\,b) &= \{x \in \mathbb{Z} : a \leq x < b\}, \\
(a\,..\,b] &= \{x \in \mathbb{Z} : a < x \leq b\}, \quad \text{and} \\
(a\,..\,b) &= \{x \in \mathbb{Z} : a < x < b\}.
\end{aligned}$$

- In this notation, a and b indicate the endpoints of the range for the set, and the type of brackets used (i.e., parenthesis versus square bracket) indicates whether each endpoint is included in the set.
- For example:
  - $[0\,..\,4]$ denotes the set of integers $\{0,1,2,3,4\}$;
  - $[0\,..\,4)$ denotes the set of integers $\{0,1,2,3\}$; and
  - $[0\,..\,N-1]$ and $[0\,..\,N)$ both denote the set of integers $\{0,1,2,\ldots,N-1\}$.

Notation for Intervals on the Real Line

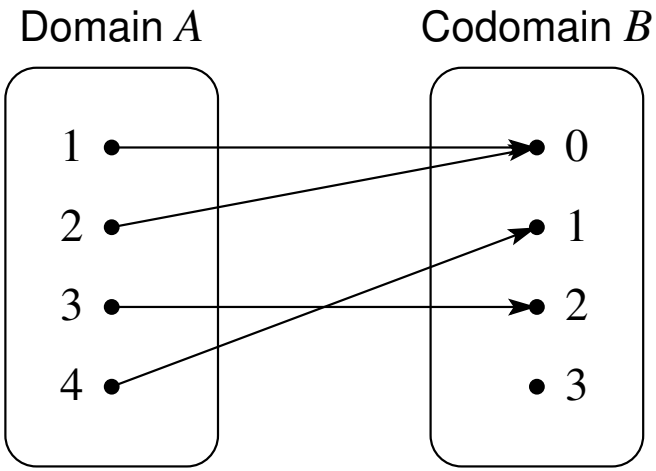
- For two real numbers a and b, we define the following notation for intervals on the real line:

$$
\begin{aligned}
[a,b] &= \{x \in \mathbb{R} : a \leq x \leq b\}, \\
(a,b) &= \{x \in \mathbb{R} : a < x < b\}, \\
[a,b) &= \{x \in \mathbb{R} : a \leq x < b\}, \quad \text{and} \\
(a,b] &= \{x \in \mathbb{R} : a < x \leq b\}.
\end{aligned}
$$

- In this notation, a and b indicate the endpoints of the interval for the set, and the type of brackets used (i.e., parenthesis versus square bracket) indicate whether each endpoint is included in the set.
- For example:
  - $[0, 100]$ denotes the set of all real numbers from 0 to 100, including both 0 and 100;
  - $(-\pi, \pi]$ denotes the set of all real numbers from $-\pi$ to π, excluding $-\pi$ but including π; and
  - $[-\pi, \pi)$ denotes the set of all real numbers from $-\pi$ to π, including $-\pi$ but excluding π.

Mappings

- A **mapping** is a relationship involving two sets that associates each element in one set, called the **domain**, with an element from the other set, called the **codomain**.
- The notation $f : A \to B$ denotes a mapping f whose domain is the set A and whose codomain is the set B.
- Example:



$$f : A \to B$$

$$A = \{1, 2, 3, 4\}$$

$$B = \{0, 1, 2, 3\}$$

$$f(x) = \begin{cases} 0 & x \in \{1, 2\} \\ 1 & x = 4 \\ 2 & x = 3. \end{cases}$$

- Although many types of mappings exist, the types of most relevance to our study of signals and systems are: functions, sequences, system operators, and transforms.

Functions

- A **function** is a mapping where the domain is a set that is ***continuous*** in nature, such as the real numbers or complex numbers.
- In practice, the codomain is typically either the real numbers or complex numbers.
- Functions are also commonly referred to as **continuous-time (CT) signals**.
- Example:
  - Let $f : \mathbb{R} \to \mathbb{R}$ such that $f(t) = t^2$ (i.e., f is the squaring function).
  - The function f maps each real number t to the real number $f(t) = t^2$.
  - The domain and codomain are the real numbers.
  - Note that f is a ***function***, whereas $f(t)$ is a ***number*** (namely, the value of the function f evaluated at t).
- Herein, we will focus almost exclusively on functions of a single independent variable (i.e., one-dimensional functions).

Sequences

- A **sequence** is a mapping where the domain is a set that is ***discrete*** in nature, such as the integers, or a subset thereof.
- In practice, the codomain is typically either the real numbers or complex numbers.
- Sequences are also commonly referred to as **discrete-time (DT) signals**.
- Example:
  - Let $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that $f(n) = n^2$, where $\mathbb{Z}^+$ denotes the set of (strictly) positive integers (i.e., f is the sequence of perfect squares).
  - The sequence f maps each (strictly) positive integer n to the (strictly) positive integer $f(n) = n^2$.
  - The domain and codomain are $\mathbb{Z}^+$ (i.e., the positive integers).
  - Note that f is a ***sequence***, whereas $f(n)$ is a ***number*** (namely, the value of the sequence f evaluated at n).
- As a matter of notation, the nth element of a sequence x is denoted as either $x(n)$ or x_n.
- Herein, we will focus almost exclusively on sequences with a single independent variable (i.e., one-dimensional sequences).

Remarks on Notation for Functions and Sequences

- For a real-valued function f of a real variable and an arbitrary real number t, the expression f denotes the function f itself and the expression $f(t)$ denotes the value of the function f evaluated at t.
- That is, f is a ***function*** and $f(t)$ is a ***number***.
- Unfortunately, the practice of using $f(t)$ to denote the function f is quite common, although strictly speaking this is an abuse of notation.
- In contexts where imprecise notation may lead to problems, one should be careful to clearly distinguish between a function and its value.
- For the real-valued functions f and g of a real variable and an arbitrary real number t:
  - The expression $f+g$ denotes a ***function***, namely, the function formed by adding the functions f and g.
  - The expression $f(t)+g(t)$ denotes a ***number***, namely, the sum of: 1) the value of the function f evaluated at t; and 2) the value of the function g evaluated at t.
- Similar comments as the ones made above for functions also hold in the case of sequences.

Remarks on Notation for Functions and Sequences (Continued)

- To express that two functions f and g are equal, we can write either:
  1. $f = g$; or
  2. $f(t) = g(t)$ for all t.
- Of the preceding two expressions, the first (i.e., $f = g$) is usually preferable, as it is less verbose.
- For the functions f and g and an operation $\circ$ that is defined pointwise for functions (such as addition, subtraction, multiplication, and division), the following relationship holds:

$$(f \circ g)(t) = f(t) \circ g(t).$$

- Some operations $\circ$ involving functions (such as convolution, to be discussed later) cannot be defined in a pointwise manner, in which case $(f \circ g)(t)$ is a valid mathematical expression, while $f(t) \circ g(t)$ is not.
- Again, similar comments as the ones made above for functions also hold in the case of sequences.

System Operators

- A **system operator** is a mapping used to represent a system.
- We will focus exclusively on the case of single-input single-output systems.
- A (single-input single-output) **system operator** maps a function or sequence representing the input of a system to a function or sequence representing the output of the system.
- The domain and codomain of a system operator are sets of ***functions or sequences***, not sets of numbers.
- Example:
  - Let $\mathcal{H} : F \to F$ such that $\mathcal{H}x(t) = 2x(t)$ (for all $t \in \mathbb{R}$) and F is the set of functions mapping $\mathbb{R}$ to $\mathbb{R}$.
  - The system $\mathcal{H}$ maps a function to a function.
  - In particular, the domain and codomain are each F, which is a set of functions.
  - The system $\mathcal{H}$ multiplies its input function x by a factor of 2 in order to produce its output function $\mathcal{H}x$.
  - Note that $\mathcal{H}x$ is a function, not a number.

Remarks on Operator Notation for CT Systems

- For a system operator $\mathcal{H}$ and a function x, $\mathcal{H}x$ is the function produced as the output of the system $\mathcal{H}$ when the input is the function x.
- Brackets around the operand of an operator are ***often omitted when not required*** for grouping.
- For example, for an operator $\mathcal{H}$, a function x, and a real number t, we would normally prefer to write:
  1. $\mathcal{H}x$ instead of the equivalent expression $\mathcal{H}(x)$; and
  2. $\mathcal{H}x(t)$ instead of the equivalent expression $\mathcal{H}(x)(t)$.
- Also, note that $\mathcal{H}x$ is a ***function*** and $\mathcal{H}x(t)$ is a ***number*** (namely, the value of the function $\mathcal{H}x$ evaluated at t).
- In the expression $\mathcal{H}(x_1+x_2)$, the brackets are needed for grouping, since $\mathcal{H}(x_1+x_2) \not\equiv \mathcal{H}x_1+x_2$ (where "$\not\equiv$" means "not equivalent").
- When multiple operators are applied, they group from ***right to left***.
- For example, for the operators $\mathcal{H}_1$ and $\mathcal{H}_2$, and the function x, the expression $\mathcal{H}_2\mathcal{H}_1x$ means $\mathcal{H}_2[\mathcal{H}_1(x)]$.

Remarks on Operator Notation for DT Systems

- For a system operator $\mathcal{H}$ and a sequence x, $\mathcal{H}x$ is the sequence produced as the output of the system $\mathcal{H}$ when the input is the sequence x.
- Brackets around the operand of an operator are ***often omitted when not required*** for grouping.
- For example, for an operator $\mathcal{H}$, a sequence x, and an integer n, we would normally prefer to write:
  1. $\mathcal{H}x$ instead of the equivalent expression $\mathcal{H}(x)$; and
  2. $\mathcal{H}x(n)$ instead of the equivalent expression $\mathcal{H}(x)(n)$.
- Also, note that $\mathcal{H}x$ is a ***sequence*** and $\mathcal{H}x(n)$ is a ***number*** (namely, the value of the sequence $\mathcal{H}x$ evaluated at n).
- In the expression $\mathcal{H}(x_1+x_2)$, the brackets are needed for grouping, since $\mathcal{H}(x_1+x_2) \not\equiv \mathcal{H}x_1+x_2$ (where "$\not\equiv$" means "not equivalent").
- When multiple operators are applied, they group from ***right to left***.
- For example, for the operators $\mathcal{H}_1$ and $\mathcal{H}_2$, and the sequence x, the expression $\mathcal{H}_2\mathcal{H}_1x$ means $\mathcal{H}_2[\mathcal{H}_1(x)]$.

Transforms

- Later, we will be introduced to several types of mappings known as transforms.
- Transforms have a mathematical structure similar to system operators.
- That is, transforms map functions/sequences to functions/sequences.
- Due to this similar structure, many of the earlier comments about system operators also apply to the case of transforms.
- For example, the Fourier transform (introduced later) is denoted as $\mathcal{F}$ and the result of applying the Fourier transform operator to the function/sequence x is denoted as $\mathcal{F}x$.
- Some examples of transforms of interest in the study of signals and systems are listed on the next slide.

Examples of Transforms

Name	Domain	Codomain
CT Fourier Series	T-periodic functions (with domain $\mathbb{R}$)	sequences (with domain $\mathbb{Z}$)
CT Fourier Transform	functions (with domain $\mathbb{R}$)	functions (with domain $\mathbb{R}$)
Laplace Transform	functions (with domain $\mathbb{R}$)	functions (with domain $\mathbb{C}$)
DT Fourier Series	N-periodic sequences (with domain $\mathbb{Z}$)	N-periodic sequences (with domain $\mathbb{Z}$)
DT Fourier Transform	sequences (with domain $\mathbb{Z}$)	2π-periodic functions (with domain $\mathbb{R}$)
Z Transform	sequences (with domain $\mathbb{Z}$)	functions (with domain $\mathbb{C}$)

Section 2.2

Properties of Signals

Even Symmetry

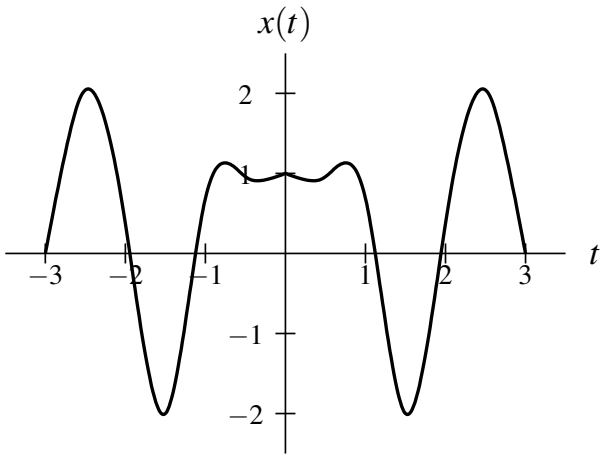
- A function x is said to be **even** if it satisfies

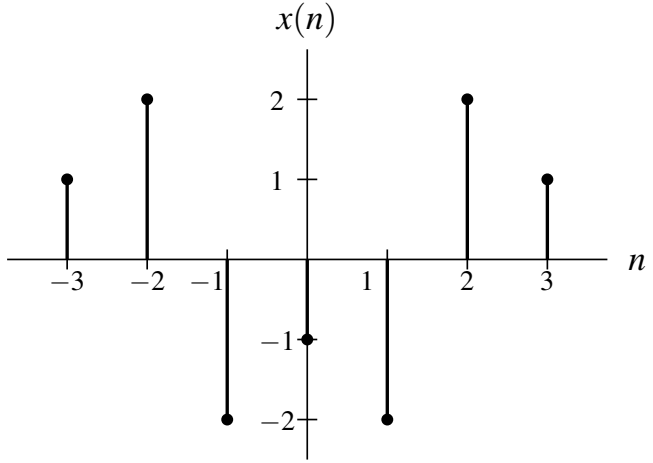
$$x(t) = x(-t) \quad \text{for all } t \text{ (where } t \text{ is a real number).}$$

- A sequence x is said to be **even** if it satisfies

$$x(n) = x(-n) \quad \text{for all } n \text{ (where } n \text{ is an integer).}$$

- Geometrically, the graph of an even signal is ***symmetric*** with respect to the vertical axis.
- Some examples of even signals are shown below.





Odd Symmetry

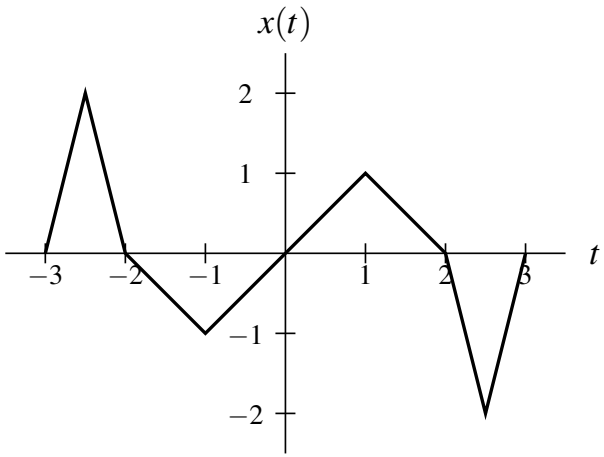
- A function x is said to be **odd** if it satisfies

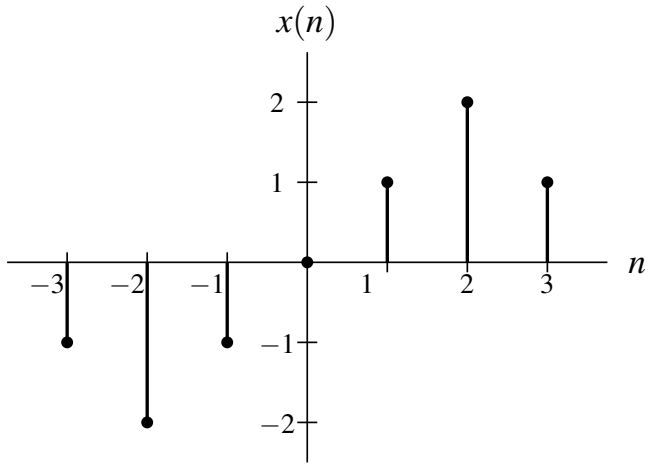
$$x(t) = -x(-t) \quad \text{for all } t \text{ (where } t \text{ is a real number).}$$

- A sequence x is said to be **odd** if it satisfies

$$x(n) = -x(-n) \quad \text{for all } n \text{ (where } n \text{ is an integer).}$$

- An odd signal x must be such that $x(0) = 0$.
- Geometrically, the graph of an odd signal is ***symmetric*** with respect to the origin.
- Some examples of odd signals are shown below.





Conjugate Symmetry

- A function x is said to be **conjugate symmetric** if it satisfies

$$x(t) = x^*(-t) \quad \text{for all } t \text{ (where } t \text{ is a real number).}$$

- A sequence x is said to be **conjugate symmetric** if it satisfies

$$x(n) = x^*(-n) \quad \text{for all } n \text{ (where } n \text{ is an integer).}$$

- The real part of a conjugate symmetric function or sequence is even.
- The imaginary part of a conjugate symmetric function or sequence is odd.
- An example of a conjugate symmetric function is a complex sinusoid $x(t) = \cos(\omega t) + j\sin(\omega t)$, where ω is a real constant.

Periodicity

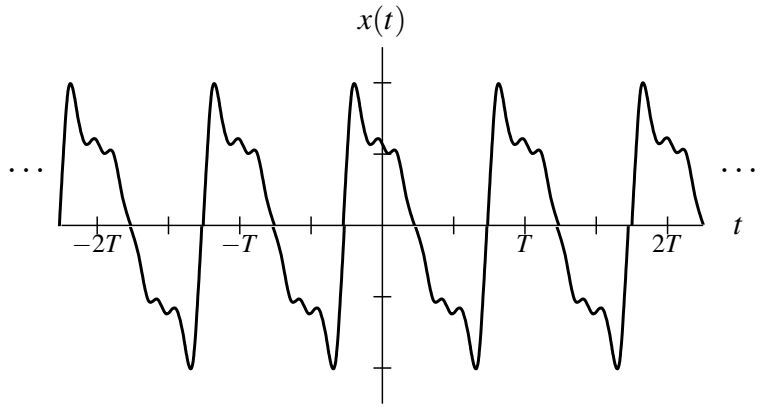
- A function x is said to be **periodic** with **period** T (or **T-periodic**) if, for some strictly-positive real constant T, the following condition holds:

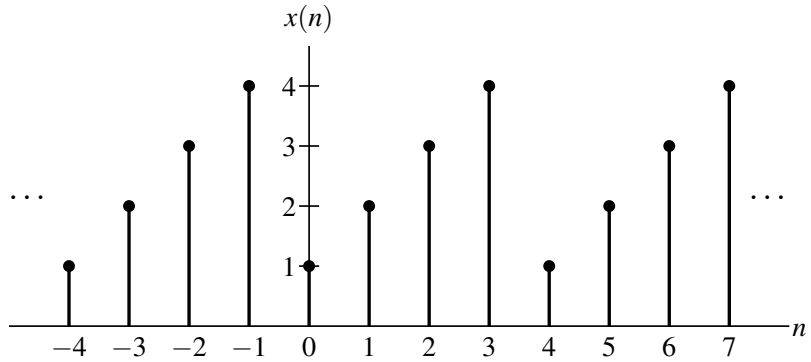
$$x(t) = x(t+T) \quad \text{for all } t \text{ (where } t \text{ is a real number).}$$

- A sequence x is said to be **periodic** with **period** N (or **N-periodic**) if, for some strictly-positive integer constant N, the following condition holds:

$$x(n) = x(n+N) \quad \text{for all } n \text{ (where } n \text{ is an integer).}$$

- Some examples of periodic signals are shown below.



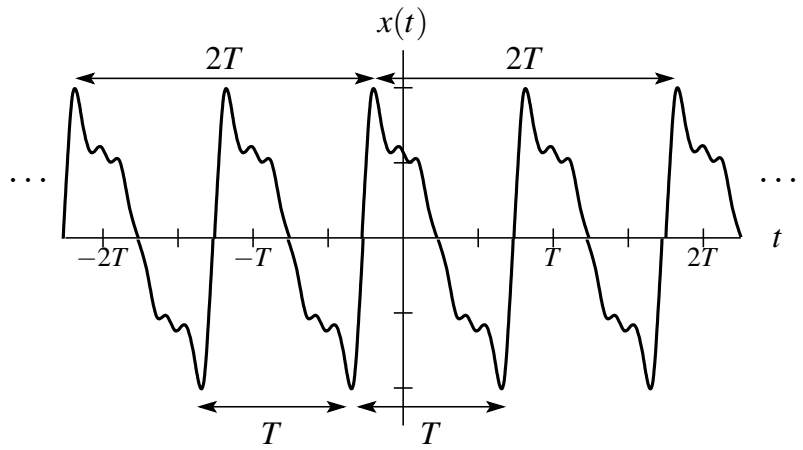


Periodicity (Continued 1)

- A function/sequence that is not periodic is said to be **aperiodic**.
- A T-periodic function x is said to have **frequency** $\frac{1}{T}$ and **angular frequency** $\frac{2\pi}{T}$.
- An N-periodic sequence x is said to have **frequency** $\frac{1}{N}$ and **angular frequency** $\frac{2\pi}{N}$.

Periodicity (Continued 2)

- The period of a periodic signal is ***not unique***. That is, a signal that is periodic with period T is also periodic with period kT, for every (strictly) positive integer k.



- The smallest period with which a signal is periodic is called the **fundamental period** and its corresponding frequency is called the **fundamental frequency**.

Part 3

Continuous-Time (CT) Signals and Systems

Section 3.1

Independent- and Dependent-Variable Transformations

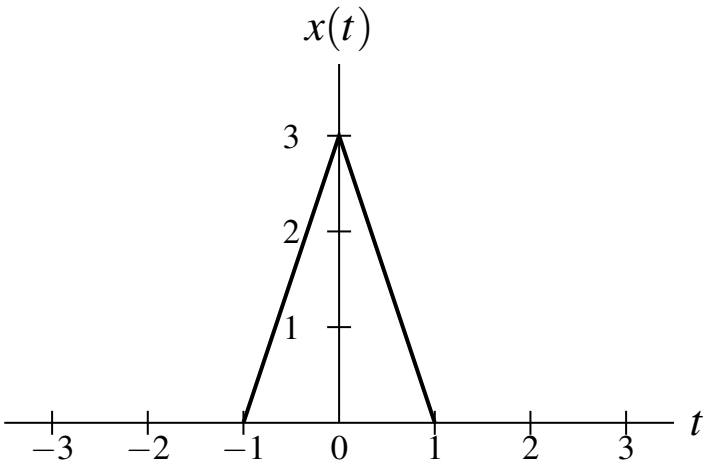
Time Shifting (Translation)

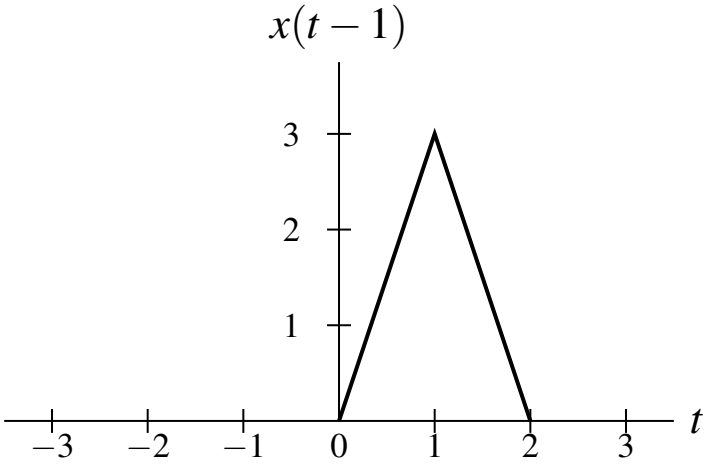
- **Time shifting** (also called **translation**) maps the input function x to the output function y as given by

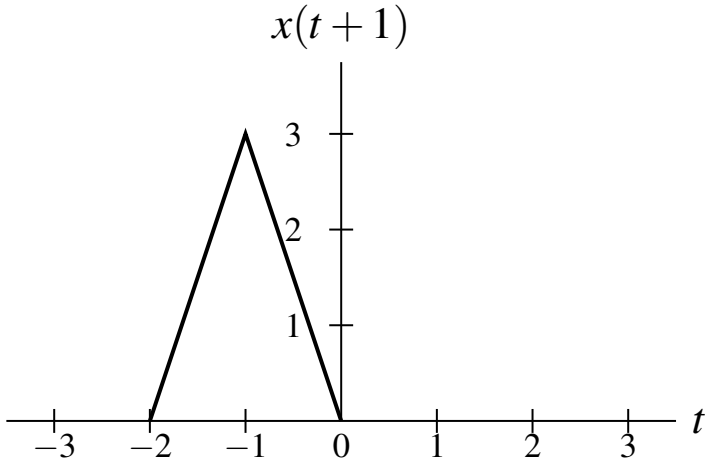
$$y(t) = x(t - b),$$

  where b is a real number.
- Such a transformation shifts the function (to the left or right) along the time axis.
- If $b > 0$, y is ***shifted to the right*** by $|b|$, relative to x (i.e., delayed in time).
- If $b < 0$, y is ***shifted to the left*** by $|b|$, relative to x (i.e., advanced in time).

Time Shifting (Translation): Example





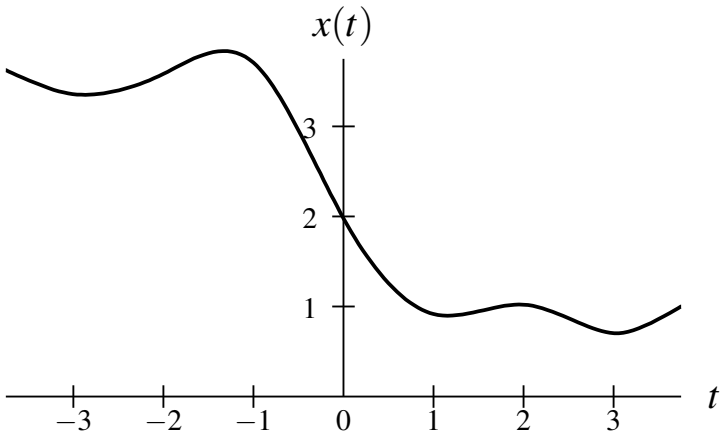


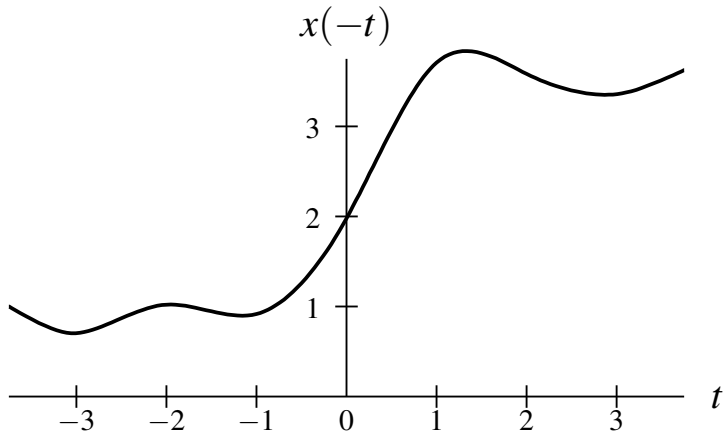
Time Reversal (Reflection)

- **Time reversal** (also known as **reflection**) maps the input function x to the output function y as given by

$$y(t) = x(-t).$$

- Geometrically, the output function y is a reflection of the input function x about the (vertical) line $t = 0$.





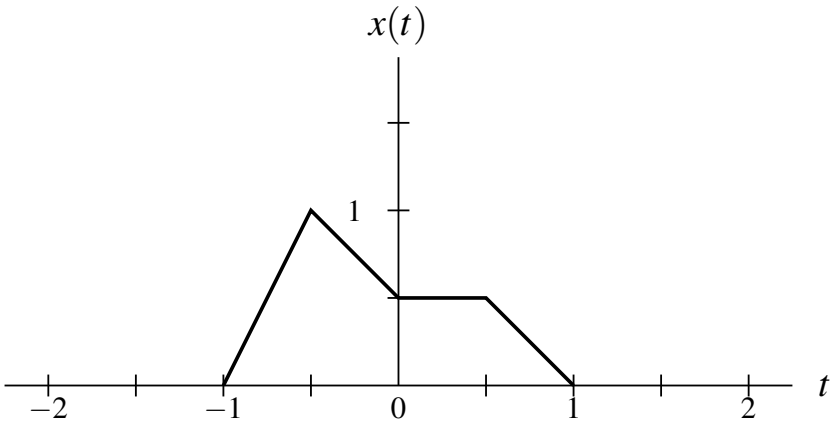
Time Compression/Expansion (Dilation)

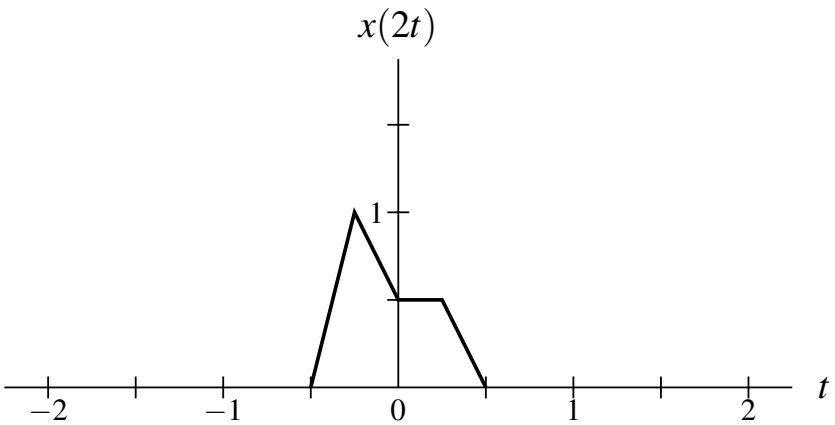
- **Time compression/expansion** (also called **dilation**) maps the input function x to the output function y as given by

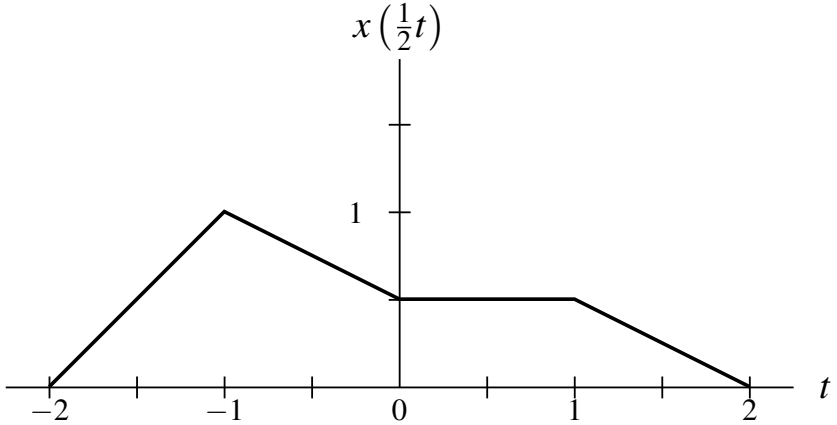
$$y(t) = x(at),$$

  where a is a ***strictly positive*** real number.
- Such a transformation is associated with a compression/expansion along the time axis.
- If $a > 1$, y is ***compressed*** along the horizontal axis by a factor of a, relative to x.
- If $a < 1$, y is ***expanded*** (i.e., stretched) along the horizontal axis by a factor of $\frac{1}{a}$, relative to x.

Time Compression/Expansion (Dilation): Example







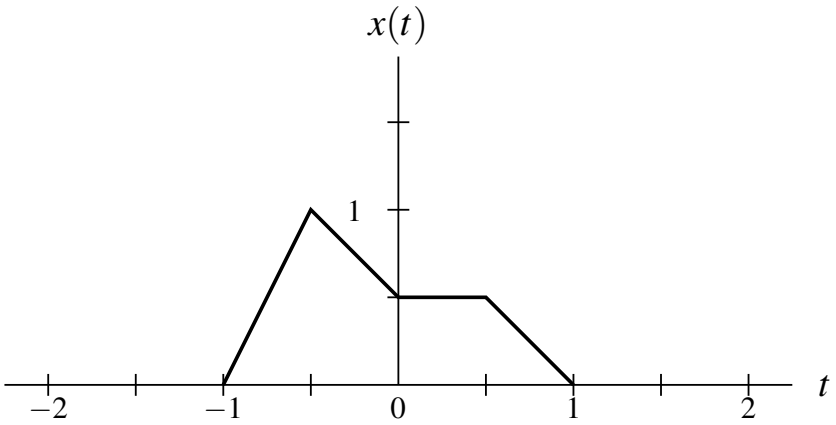
Time Scaling (Dilation/Reflection)

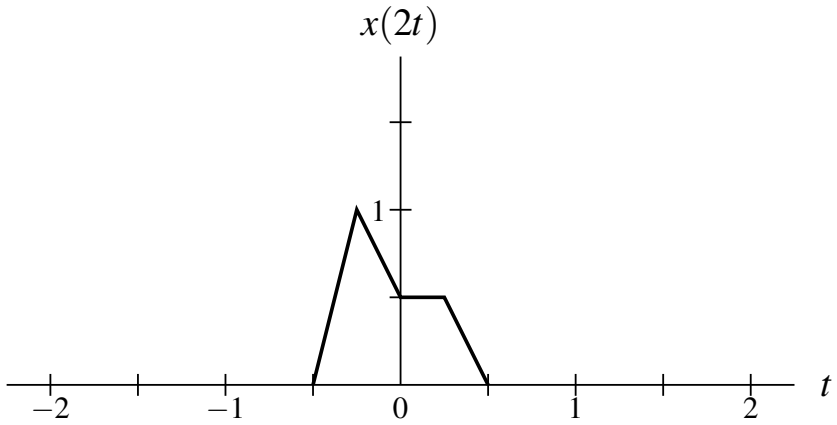
- **Time scaling** maps the input function x to the output function y as given by

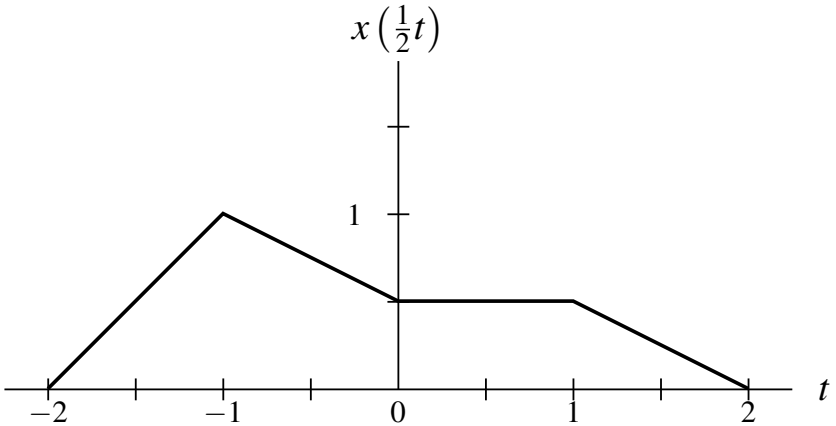
$$y(t) = x(at),$$

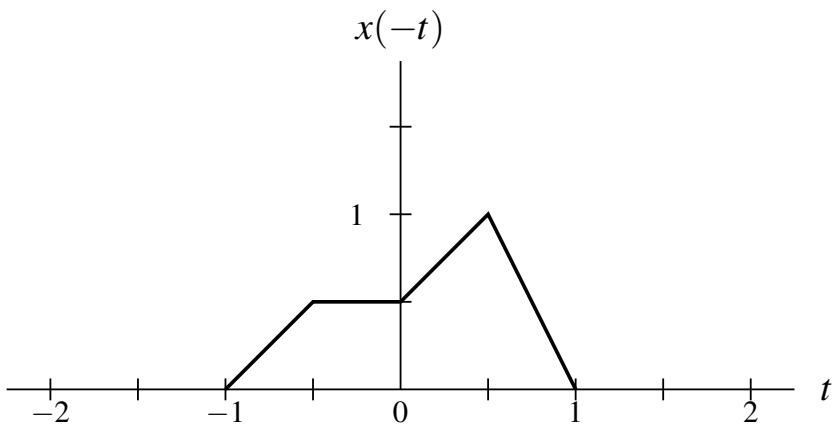
  where a is a ***nonzero*** real number.
- Such a transformation is associated with a dilation (i.e., compression/expansion along the time axis) and/or time reversal.
- If $|a| > 1$, the function is ***compressed*** along the time axis by a factor of $|a|$.
- If $|a| < 1$, the function is ***expanded*** (i.e., stretched) along the time axis by a factor of $\left|\frac{1}{a}\right|$.
- If $|a| = 1$, the function is neither expanded nor compressed.
- If $a < 0$, the function is also time reversed.
- Dilation (i.e., expansion/compression) and time reversal ***commute***.
- Time reversal is a special case of time scaling with $a = -1$; and time compression/expansion is a special case of time scaling with $a > 0$.

Time Scaling (Dilation/Reflection): Example







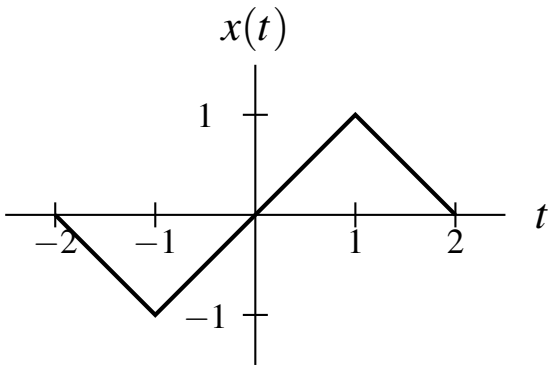


Combined Time Scaling and Time Shifting

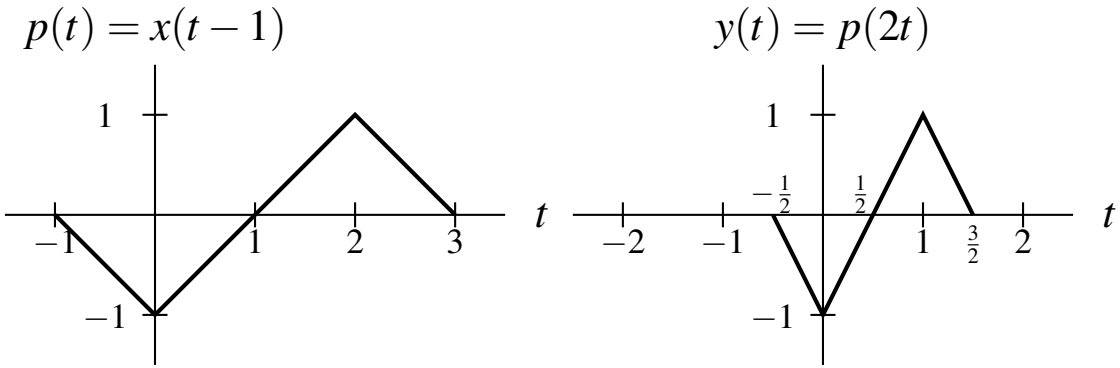
- Consider a transformation that maps the input function x to the output function y as given by
$$y(t) = x(at - b),$$
where a and b are real numbers and $a \neq 0$.
- The above transformation can be shown to be the combination of a time-scaling operation and time-shifting operation.
- Since time scaling and time shifting ***do not commute***, we must be particularly careful about the order in which these transformations are applied.
- The above transformation has two distinct but equivalent interpretations:
  1. first, time shifting x by b, and then time scaling the result by a;
  2. first, time scaling x by a, and then time shifting the result by b/a.
- Note that the time shift is not by the same amount in both cases.
- In particular, note that when time scaling is applied first followed by time shifting, the time shift is by b/a, not b.

Combined Time Scaling and Time Shifting: Example

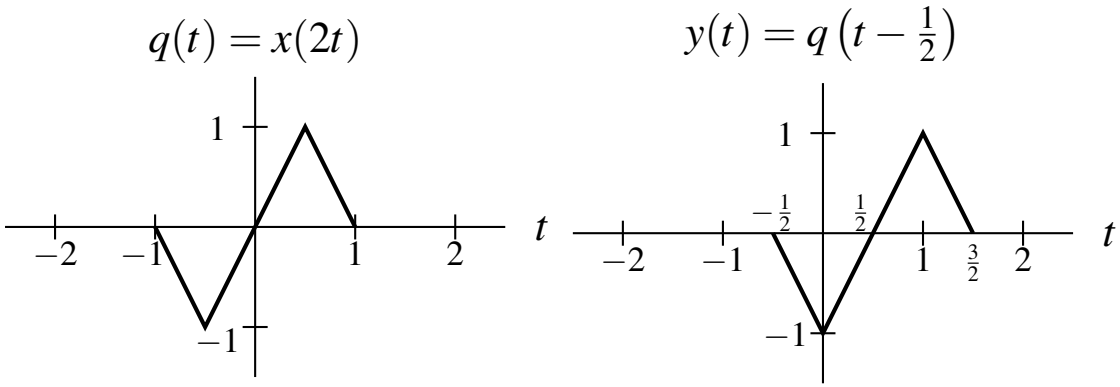
Given x as shown below, find $y(t) = x(2t-1)$.



time shift by 1 and then time scale by 2



time scale by 2 and then time shift by $\frac{1}{2}$



Two Perspectives on Independent-Variable Transformations

- A transformation of the independent variable can be viewed in terms of
  1. the effect that the transformation has on the ***function***; or
  2. the effect that the transformation has on the ***horizontal axis***.
- This distinction is important because such a transformation has ***opposite*** effects on the function and horizontal axis.
- For example, the (time-shifting) transformation that replaces t by $t-b$ (where b is a real number) in $x(t)$ can be viewed as a transformation that
  1. shifts the function x ***right*** by b units; or
  2. shifts the horizontal axis ***left*** by b units.
- In our treatment of independent-variable transformations, we are only interested in the effect that a transformation has on the ***function***.
- If one is not careful to consider that we are interested in the function perspective (as opposed to the axis perspective), many aspects of independent-variable transformations will not make sense.

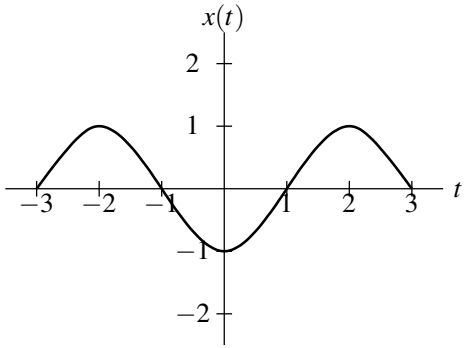
Amplitude Scaling

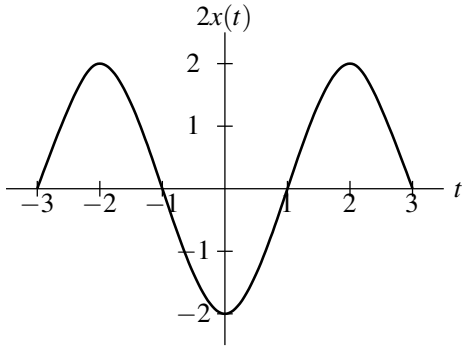
- **Amplitude scaling** maps the input function x to the output function y as given by

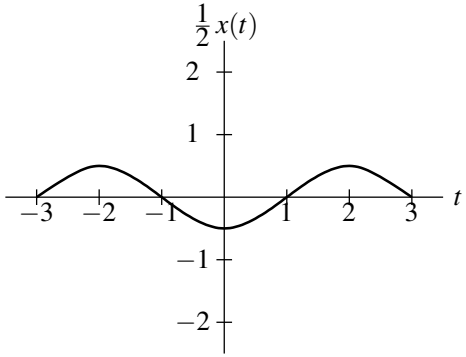
$$y(t) = ax(t),$$

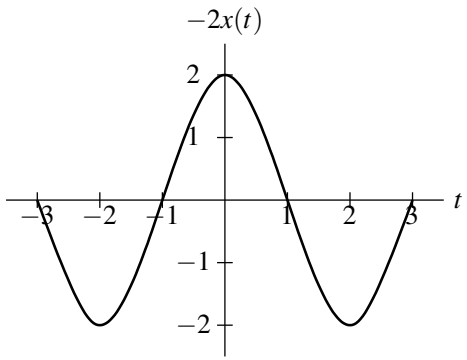
where a is a real number.

- Geometrically, the output function y is ***expanded/compressed*** in amplitude and/or ***reflected*** about the horizontal axis.









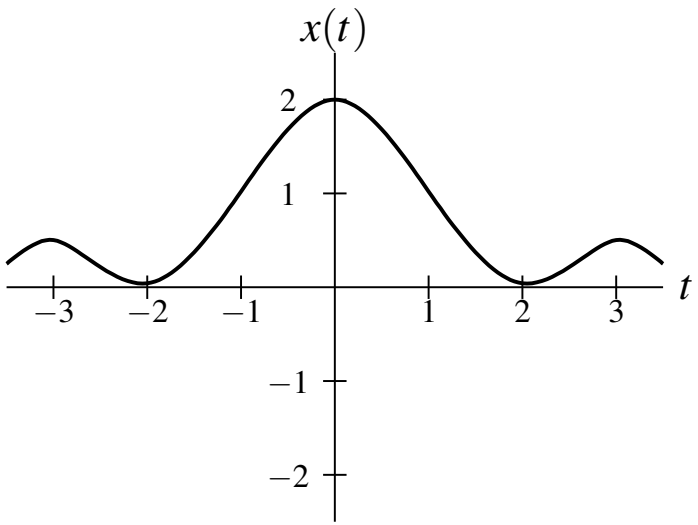
Amplitude Shifting

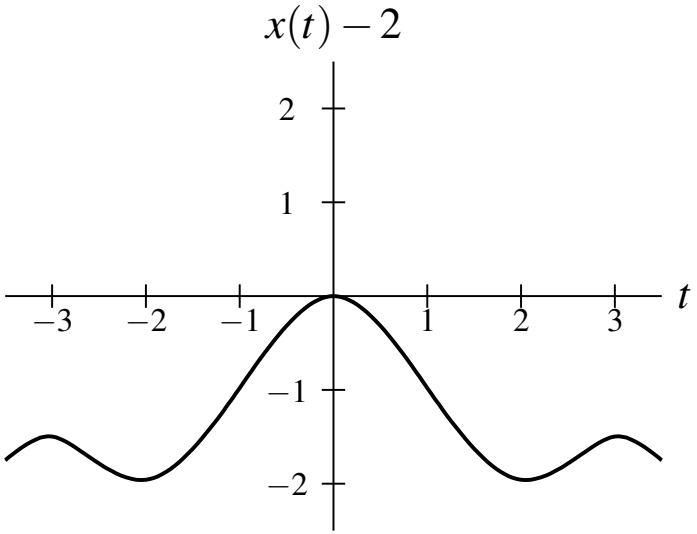
- **Amplitude shifting** maps the input function x to the output function y as given by

$$y(t) = x(t) + b,$$

where b is a real number.

- Geometrically, amplitude shifting adds a ***vertical displacement*** to x.





Combined Amplitude Scaling and Amplitude Shifting

- We can also combine amplitude scaling and amplitude shifting transformations.
- Consider a transformation that maps the input function x to the output function y, as given by

$$y(t) = ax(t) + b,$$

  where a and b are real numbers.
- Equivalently, the above transformation can be expressed as

$$y(t) = a\left[x(t) + \frac{b}{a}\right].$$

- The above transformation is equivalent to:
  1. first amplitude scaling x by a, and then amplitude shifting the resulting function by b; or
  2. first amplitude shifting x by b/a, and then amplitude scaling the resulting function by a.

Section 3.2

Properties of Functions

Symmetry and Addition/Multiplication

- Sums involving even and odd functions have the following properties:
  - The sum of two even functions is even.
  - The sum of two odd functions is odd.
  - The sum of an even function and odd function is neither even nor odd, provided that neither of the functions is identically zero.
- That is, the ***sum*** of functions with the ***same type of symmetry*** also has the ***same type of symmetry***.
- Products involving even and odd functions have the following properties:
  - The product of two even functions is even.
  - The product of two odd functions is even.
  - The product of an even function and an odd function is odd.
- That is, the ***product*** of functions with the ***same type of symmetry*** is ***even***, while the ***product*** of functions with ***opposite types of symmetry*** is ***odd***.

Decomposition of a Function into Even and Odd Parts

- Every function x has a ***unique*** representation of the form

$$x(t) = x_{\mathrm{e}}(t) + x_{\mathrm{o}}(t),$$

  where the functions x_{e} and x_{o} are ***even*** and ***odd***, respectively.
- In particular, the functions x_{e} and x_{o} are given by

$$x_{\mathrm{e}}(t) = \tfrac{1}{2}\left[x(t) + x(-t)\right] \quad \text{and} \quad x_{\mathrm{o}}(t) = \tfrac{1}{2}\left[x(t) - x(-t)\right].$$

- The functions x_{e} and x_{o} are called the **even part** and **odd part** of x, respectively.
- For convenience, the even and odd parts of x are often denoted as $\mathrm{Even}\{x\}$ and $\mathrm{Odd}\{x\}$, respectively.

Sum of Periodic Functions

- **Sum of periodic functions.** For two periodic functions x_1 and x_2 with fundamental periods T_1 and T_2, respectively, and the sum $y = x_1 + x_2$:
  1. The sum y is periodic if and only if the ratio T_1/T_2 is a ***rational number*** (i.e., the quotient of two integers).
  2. If y is periodic, its fundamental period is rT_1 (or equivalently, qT_2, since $rT_1 = qT_2$), where $T_1/T_2 = q/r$ and q and r are integers and ***coprime*** (i.e., have no common factors). (Note that rT_1 is simply the least common multiple of T_1 and T_2.)
- Although the above theorem only directly addresses the case of the sum of two functions, the case of N functions (where $N > 2$) can be handled by applying the theorem repeatedly $N - 1$ times.

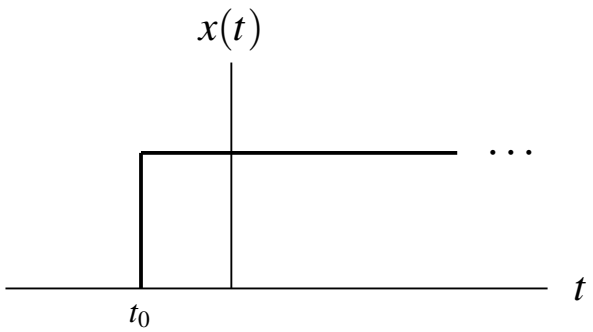
Right-Sided Functions

- A function x is said to be **right sided** if, for some (finite) real constant t_0, the following condition holds:

$$x(t) = 0 \quad \text{for all } t < t_0$$

(i.e., x is ***only potentially nonzero to the right of*** t_0).

- An example of a right-sided function is shown below.



- A function x is said to be **causal** if

$$x(t) = 0 \quad \text{for all } t < 0.$$

- A causal function is a ***special case*** of a right-sided function.
- A causal function is not to be confused with a causal system. In these two contexts, the word "causal" has very different meanings.

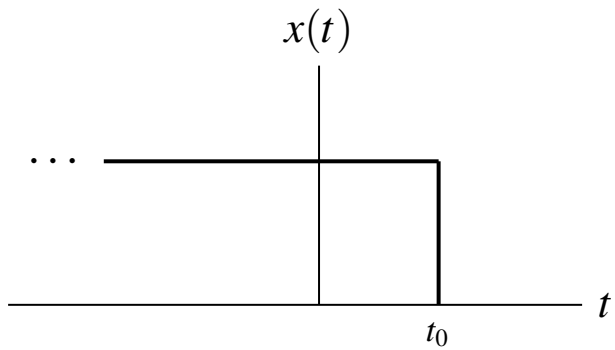
Left-Sided Functions

- A function x is said to be **left sided** if, for some (finite) real constant t_0, the following condition holds:

$$x(t) = 0 \quad \text{for all } t > t_0$$

(i.e., x is ***only potentially nonzero to the left of*** t_0).

- An example of a left-sided function is shown below.



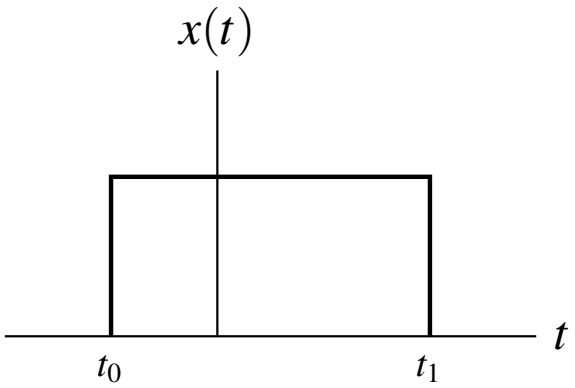
- Similarly, a function x is said to be **anticausal** if

$$x(t) = 0 \quad \text{for all } t > 0.$$

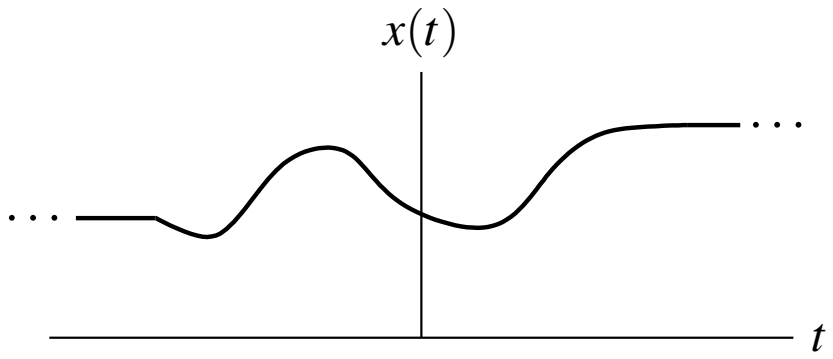
- An anticausal function is a ***special case*** of a left-sided function.
- An anticausal function is not to be confused with an anticausal system. In these two contexts, the word "anticausal" has very different meanings.

Finite-Duration and Two-Sided Functions

- A function that is both left sided and right sided is said to be **finite duration** (or **time limited**).
- An example of a finite duration function is shown below.



- A function that is neither left sided nor right sided is said to be **two sided**.
- An example of a two-sided function is shown below.



Bounded Functions

- A function x is said to be **bounded** if there exists some (***finite***) positive real constant A such that

$$|x(t)| \leq A \quad \text{for all } t$$

(i.e., $x(t)$ is ***finite*** for all t).

- For example, the sine and cosine functions are bounded, since

$$|\sin t| \leq 1 \text{ for all } t \quad \text{and} \quad |\cos t| \leq 1 \text{ for all } t.$$

- In contrast, the tangent function and any nonconstant polynomial function p (e.g., $p(t) = t^2$) are unbounded, since

$$\lim_{t \to \pi/2} |\tan t| = \infty \quad \text{and} \quad \lim_{|t| \to \infty} |p(t)| = \infty.$$

Energy and Power of a Function

- The **energy** E contained in the function x is given by

$$E = \int_{-\infty}^{\infty} |x(t)|^2 \, dt.$$

- A signal with finite energy is said to be an **energy signal**.
- The **average power** P contained in the function x is given by

$$P = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} |x(t)|^2 \, dt.$$

- A signal with (nonzero) finite average power is said to be a **power signal**.

Section 3.3

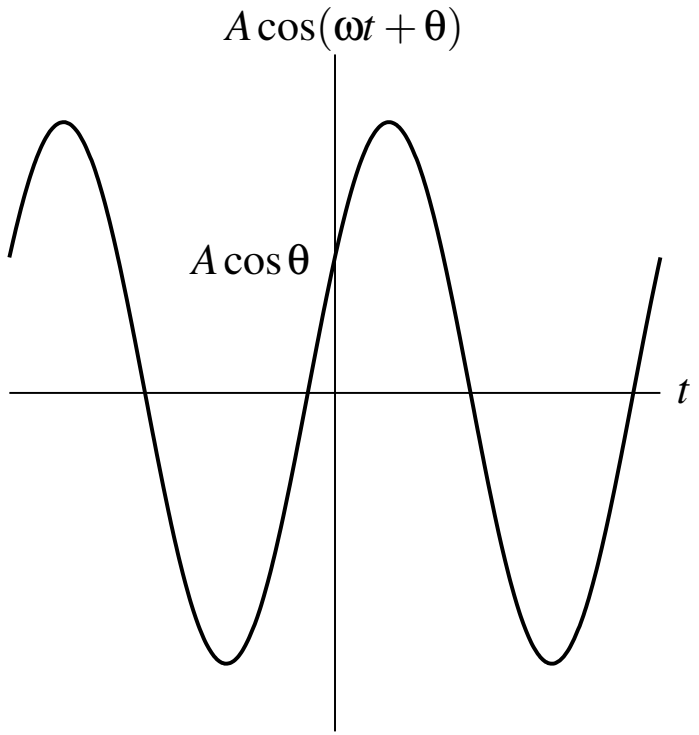
Elementary Functions

Real Sinusoidal Functions

- A **real sinusoidal function** is a function of the form

$$x(t) = A\cos(\omega t + \theta),$$

  where A, ω, and θ are ***real*** constants.
- Such a function is periodic with ***fundamental period*** $T = \frac{2\pi}{|\omega|}$ and ***fundamental frequency*** $|\omega|$.
- A real sinusoid has a plot resembling that shown below.



Complex Exponential Functions

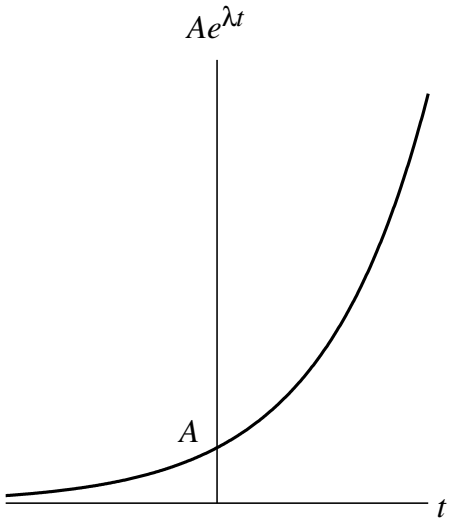
- A **complex exponential function** is a function of the form

$$x(t) = Ae^{\lambda t},$$

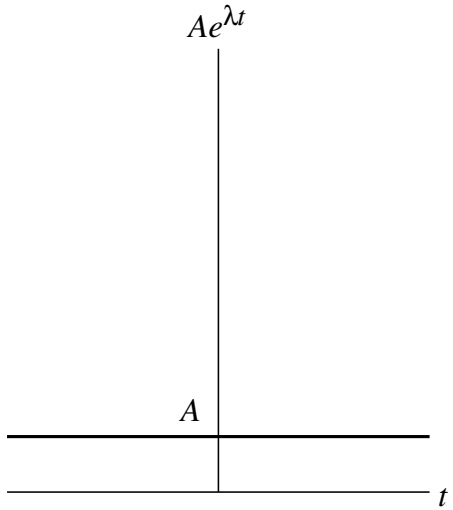
  where A and λ are ***complex*** constants.
- A complex exponential can exhibit one of a number of ***distinct modes of behavior***, depending on the values of its parameters A and λ.
- For example, as special cases, complex exponentials include real exponentials and complex sinusoids.

Real Exponential Functions

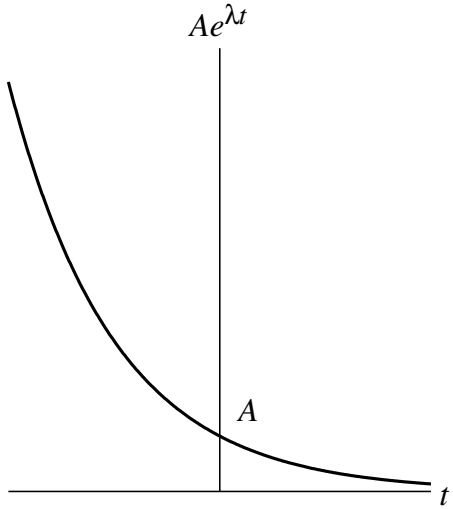
- A **real exponential function** is a special case of a complex exponential $x(t) = Ae^{\lambda t}$, where A and λ are restricted to be ***real*** numbers.
- A real exponential can exhibit one of ***three distinct modes*** of behavior, depending on the value of λ, as illustrated below.
- If $\lambda > 0$, $x(t)$ ***increases*** exponentially as t increases (i.e., a growing exponential).
- If $\lambda < 0$, $x(t)$ ***decreases*** exponentially as t increases (i.e., a decaying exponential).
- If $\lambda = 0$, $x(t)$ simply equals the ***constant*** A.



$\lambda > 0$



$\lambda = 0$



$\lambda < 0$

Complex Sinusoidal Functions

- A complex sinusoidal function is a special case of a complex exponential $x(t) = Ae^{\lambda t}$, where A is ***complex*** and λ is ***purely imaginary*** (i.e., $\mathrm{Re}\{\lambda\} = 0$).
- That is, a **complex sinusoidal function** is a function of the form

$$x(t) = Ae^{j\omega t},$$

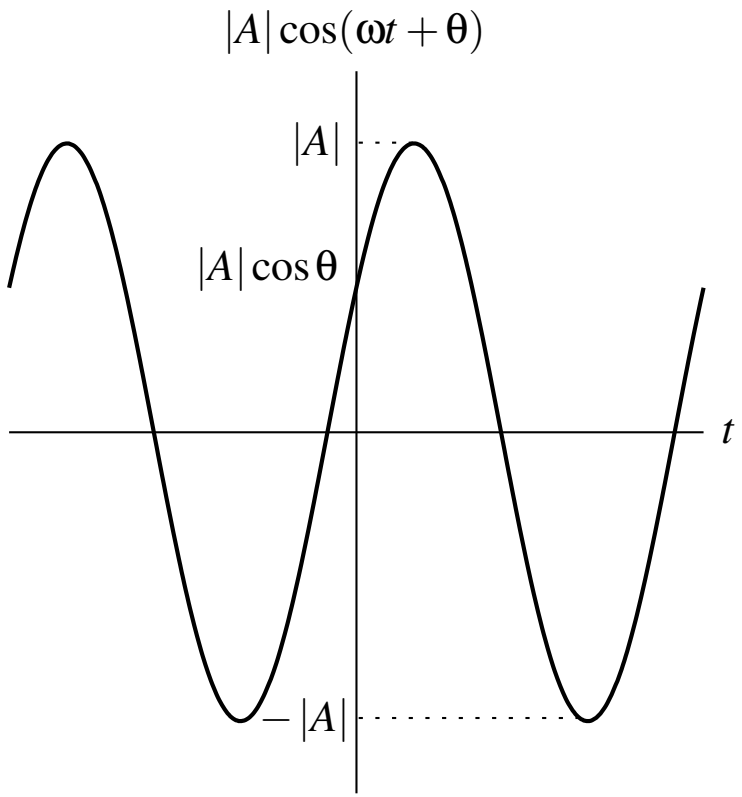
  where A is ***complex*** and ω is ***real***.
- By expressing A in polar form as $A = |A|\, e^{j\theta}$ (where θ is real) and using Euler's relation, we can rewrite $x(t)$ as

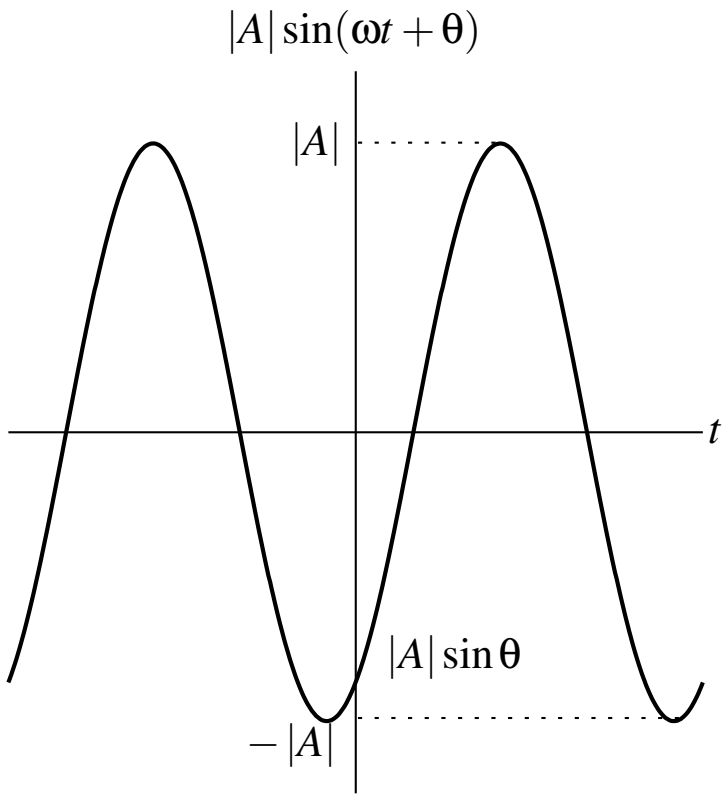
$$x(t) = \underbrace{|A|\cos(\omega t + \theta)}_{\mathrm{Re}\{x(t)\}} + j\underbrace{|A|\sin(\omega t + \theta)}_{\mathrm{Im}\{x(t)\}}.$$

- Thus, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are the same except for a time shift.
- Also, x is periodic with ***fundamental period*** $T = \frac{2\pi}{|\omega|}$ and ***fundamental frequency*** $|\omega|$.

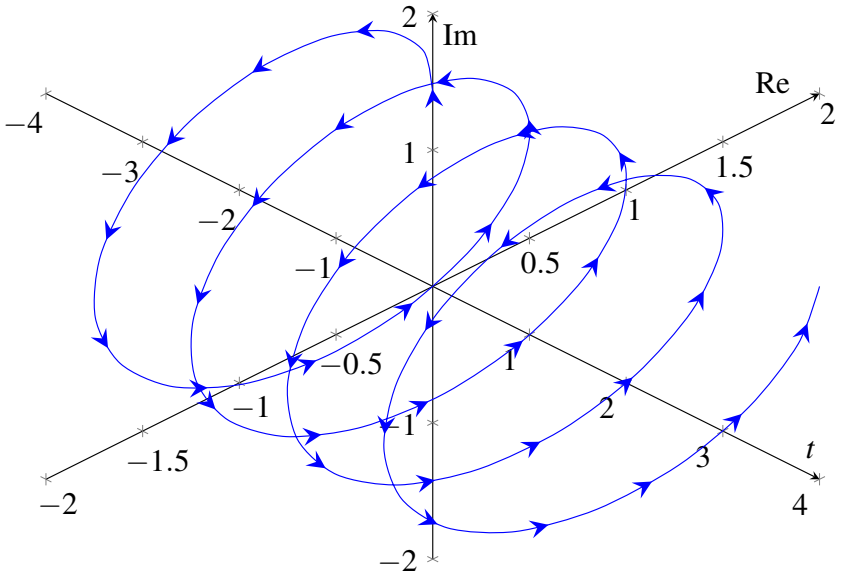
Complex Sinusoidal Functions (Continued)

- The graphs of $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ have the forms shown below.

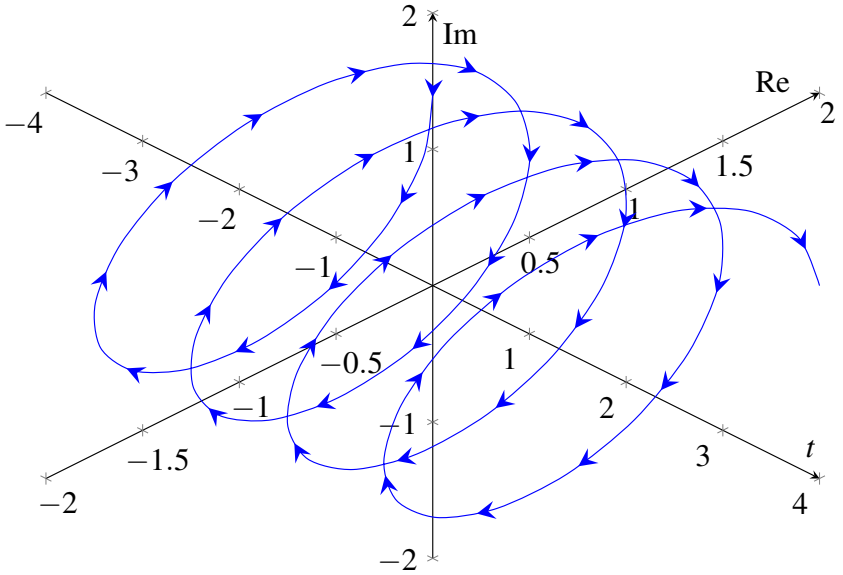




Plot of $x(t) = e^{j\omega t}$ for $\omega \in \{2\pi, -2\pi\}$



$\omega = 2\pi$



$\omega = -2\pi$

General Complex Exponential Functions

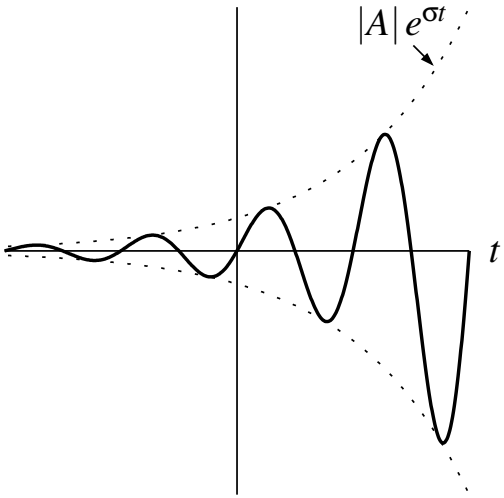
- In the most general case of a complex exponential function $x(t) = Ae^{\lambda t}$, A and λ are both ***complex***.
- Letting $A = |A|e^{j\theta}$ and $\lambda = \sigma + j\omega$ (where θ, σ, and ω are real), and using Euler's relation, we can rewrite $x(t)$ as

$$x(t) = \underbrace{|A|e^{\sigma t}\cos(\omega t + \theta)}_{\mathrm{Re}\{x(t)\}} + j\underbrace{|A|e^{\sigma t}\sin(\omega t + \theta)}_{\mathrm{Im}\{x(t)\}}.$$

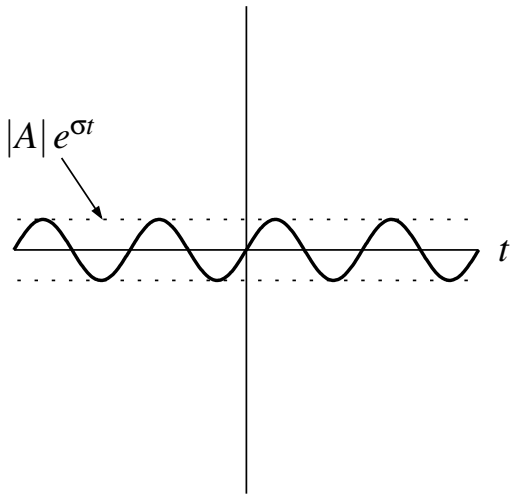
- Thus, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are each the product of a real exponential and real sinusoid.
- One of ***three distinct modes*** of behavior is exhibited by $x(t)$, depending on the value of σ.
- If $\sigma = 0$, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are ***real sinusoids***.
- If $\sigma > 0$, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are each the ***product of a real sinusoid and a growing real exponential***.
- If $\sigma < 0$, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are each the ***product of a real sinusoid and a decaying real exponential***.

General Complex Exponential Functions (Continued)

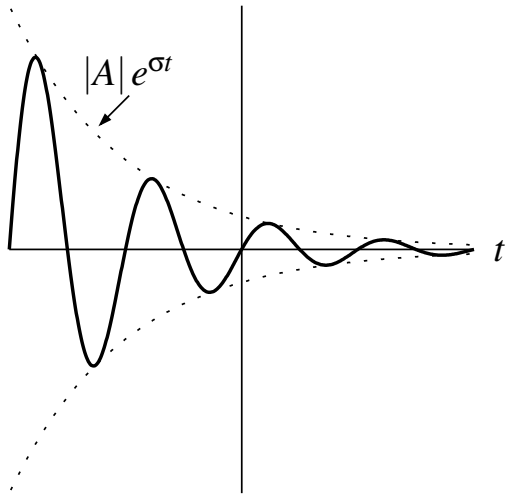
- The ***three modes of behavior*** for $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are illustrated below.



$\sigma > 0$



$\sigma = 0$



$\sigma < 0$

Relationship Between Complex Exponentials and Real Sinusoids

- From Euler's relation, a complex sinusoid can be expressed as the sum of two real sinusoids as

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

- Moreover, a real sinusoid can be expressed as the sum of two complex sinusoids using the identities

$$A\cos(\omega t + \theta) = \frac{A}{2}\left[e^{j(\omega t+\theta)} + e^{-j(\omega t+\theta)}\right] \quad \text{and}$$

$$A\sin(\omega t + \theta) = \frac{A}{2j}\left[e^{j(\omega t+\theta)} - e^{-j(\omega t+\theta)}\right].$$

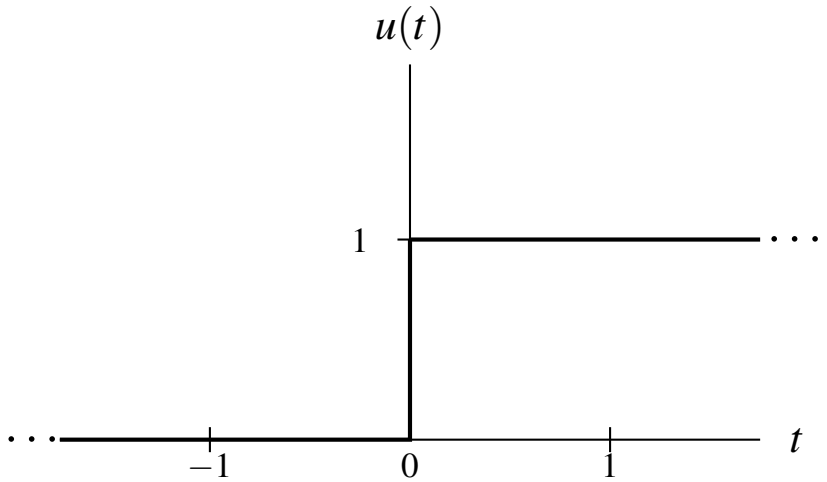
- Note that, above, we are simply ***restating results*** from the (appendix) material on complex analysis.

Unit-Step Function

- The **unit-step function** (also known as the **Heaviside function**), denoted u, is defined as

$$u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

- Due to the manner in which u is used in practice, the actual ***value of*** $u(0)$ is unimportant. Sometimes values of 0 and $\frac{1}{2}$ are also used for $u(0)$.
- A plot of this function is shown below.

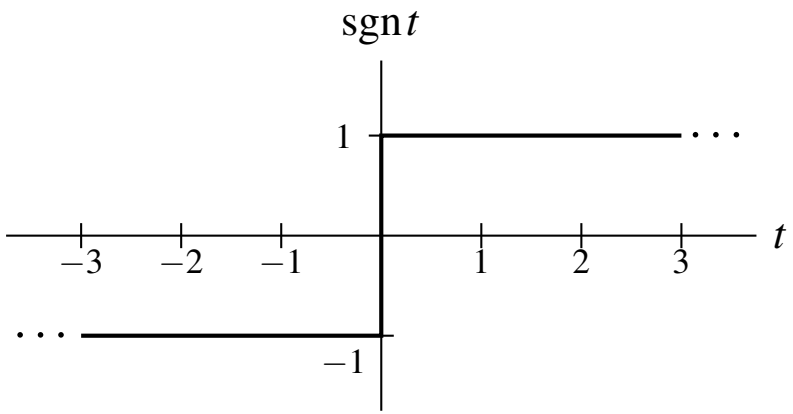


Signum Function

- The **signum function**, denoted sgn, is defined as

$$\operatorname{sgn} t = \begin{cases} 1 & t > 0 \\ 0 & t = 0 \\ -1 & t < 0. \end{cases}$$

- From its definition, one can see that the signum function simply computes the ***sign*** of a number.
- A plot of this function is shown below.

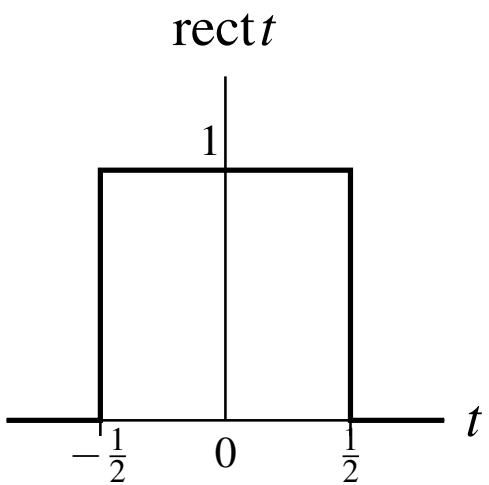


Rectangular Function

- The **rectangular function** (also called the unit-rectangular pulse function), denoted rect, is given by

$$\mathrm{rect}\,t = \begin{cases} 1 & -\frac{1}{2} \leq t < \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

- Due to the manner in which the rect function is used in practice, the actual ***value of*** $\mathrm{rect}\,t$ ***at*** $t = \pm\frac{1}{2}$ is unimportant. Sometimes different values are used from those specified above.
- A plot of this function is shown below.



Indicator Function

- Functions and sequences that are one over some subset of their domain and zero elsewhere appear very frequently in engineering (e.g., the unit-step function and rectangular function).
- Indicator function notation provides a concise way to denote such functions and sequences.
- The **indicator function** of a subset S of a set A, denoted χ_S, is defined as

$$\chi_S(t) = \begin{cases} 1 & \text{if } t \in S \\ 0 & \text{otherwise.} \end{cases}$$

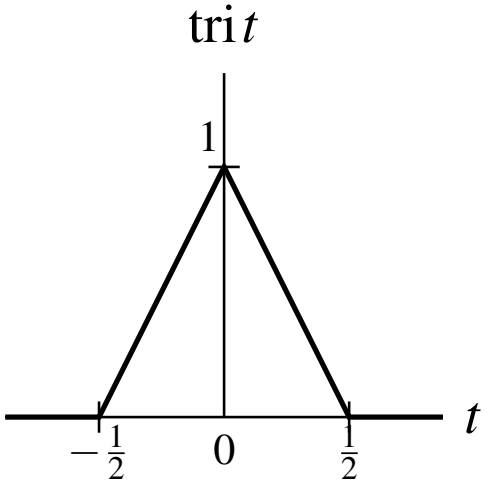
- A rectangular pulse (defined on $\mathbb{R}$) having an amplitude of 1, a leading edge at a, and falling edge at b is $\chi_{[a,b]}$.
- The unit-step function (defined on $\mathbb{R}$) is $\chi_{[0,\infty)}$.
- The unit-rectangular pulse (defined on $\mathbb{R}$) is $\chi_{[-1/2,1/2]}$.
- The unit-step sequence (defined on $\mathbb{Z}$) is $\chi_{[0..\infty)}$.

Triangular Function

- The **triangular function** (also called the unit-triangular pulse function), denoted tri, is defined as

$$\mathrm{tri}\, t = \begin{cases} 1 - 2|t| & |t| \leq \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

- A plot of this function is shown below.

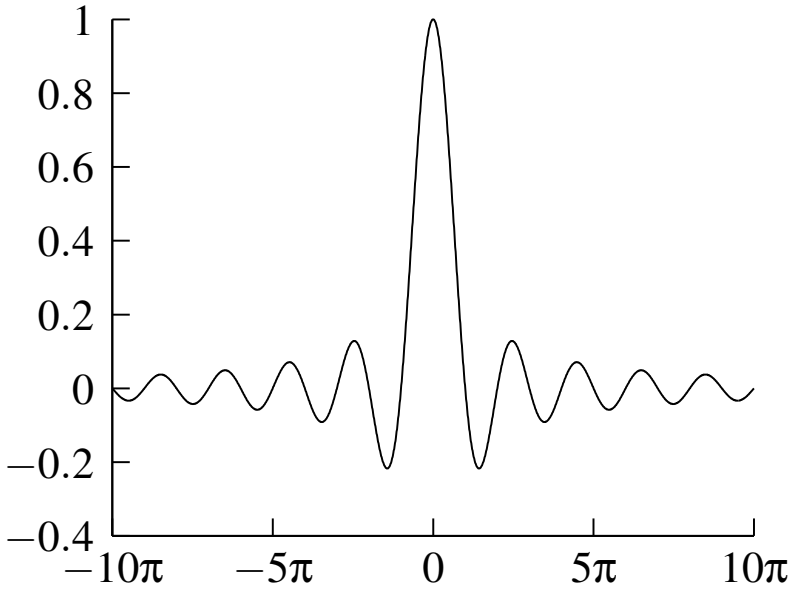


Cardinal Sine Function

- The **cardinal sine** function, denoted sinc, is given by

$$\operatorname{sinc} t = \frac{\sin t}{t}.$$

- By l'Hopital's rule, $\operatorname{sinc} 0 = 1$.
- A plot of this function for part of the real line is shown below. [Note that the oscillations in $\operatorname{sinc} t$ do not die out for finite t.]



Floor and Ceiling Functions

- The **floor function**, denoted $\lfloor \cdot \rfloor$, is a function that maps a real number x to the largest integer not more than x.
- In other words, the floor function rounds a real number to the nearest integer in the direction of negative infinity.
- For example,

$$\left\lfloor -\tfrac{1}{2} \right\rfloor = -1, \quad \left\lfloor \tfrac{1}{2} \right\rfloor = 0, \quad \text{and} \quad \lfloor 1 \rfloor = 1.$$

- The **ceiling function**, denoted $\lceil \cdot \rceil$, is a function that maps a real number x to the smallest integer not less than x.
- In other words, the ceiling function rounds a real number to the nearest integer in the direction of positive infinity.
- For example,

$$\left\lceil -\tfrac{1}{2} \right\rceil = 0, \quad \left\lceil \tfrac{1}{2} \right\rceil = 1, \quad \text{and} \quad \lceil 1 \rceil = 1.$$

Some Properties of the Floor and Ceiling Functions

- Several useful properties of the floor and ceiling functions include:

$$\lfloor x+n \rfloor = \lfloor x \rfloor + n \quad \text{for } x \in \mathbb{R} \text{ and } n \in \mathbb{Z};$$

$$\lceil x+n \rceil = \lceil x \rceil + n \quad \text{for } x \in \mathbb{R} \text{ and } n \in \mathbb{Z};$$

$$\lceil x \rceil = -\lfloor -x \rfloor \quad \text{for } x \in \mathbb{R};$$

$$\lfloor x \rfloor = -\lceil -x \rceil \quad \text{for } x \in \mathbb{R};$$

$$\left\lceil \frac{m}{n} \right\rceil = \left\lfloor \frac{m+n-1}{n} \right\rfloor = \left\lfloor \frac{m-1}{n} \right\rfloor + 1 \quad \text{for } m,n \in \mathbb{Z} \text{ and } n > 0; \quad \text{and}$$

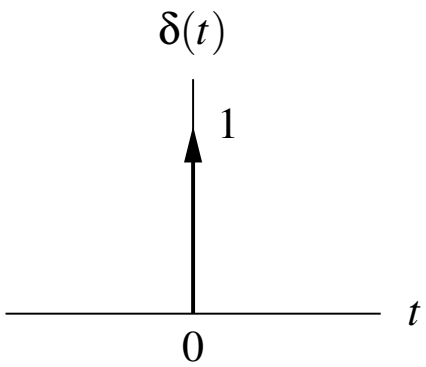
$$\left\lfloor \frac{m}{n} \right\rfloor = \left\lceil \frac{m-n+1}{n} \right\rceil = \left\lceil \frac{m+1}{n} \right\rceil - 1 \quad \text{for } m,n \in \mathbb{Z} \text{ and } n > 0.$$

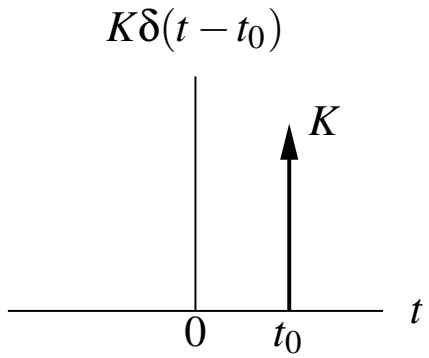
Delta Function

- The **delta function** (also known as the **Dirac delta function** or **unit-impulse function**), denoted δ, is defined as the function with the following two properties:

$$\delta(t) = 0 \quad \text{for } t \neq 0 \quad \text{and}$$
$$\int_{-\infty}^{\infty} \delta(t)dt = 1.$$

- Technically, δ is not a function in the ordinary sense. Rather, it is what is known as a ***generalized function***. Consequently, the δ function sometimes behaves in unusual ways.
- Graphically, the delta function is represented as shown below.



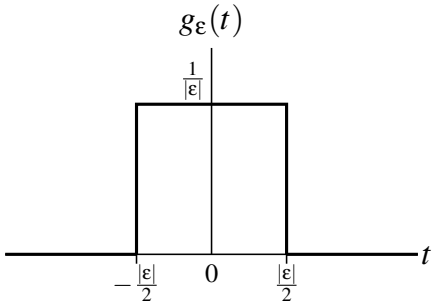


Delta Function as a Limit

- Consider the function g_ε (where ε is a real constant) defined by

$$g_\varepsilon(t) = \begin{cases} \frac{1}{|\varepsilon|} & |t| < \frac{|\varepsilon|}{2} \\ 0 & \text{otherwise.} \end{cases}$$

- A plot of g_ε is shown below.



- Clearly, for any choice of ε, $\int_{-\infty}^{\infty} g_\varepsilon(t)dt = 1$.
- The function δ can be obtained as the following limit:

$$\delta(t) = \lim_{\varepsilon \to 0} g_\varepsilon(t).$$

- That is, δ can be viewed as a ***limiting case of a rectangular pulse*** where the pulse width becomes infinitesimally small and the pulse height becomes infinitely large in such a way that the integral of the resulting function remains unity.

Properties of Delta Function

- **Equivalence property.** For any continuous function x and any real constant t_0,

$$x(t)\delta(t-t_0) = x(t_0)\delta(t-t_0).$$

- **Sifting property.** For any continuous function x and any real constant t_0,

$$\int_{-\infty}^{\infty} x(t)\delta(t-t_0)dt = x(t_0).$$

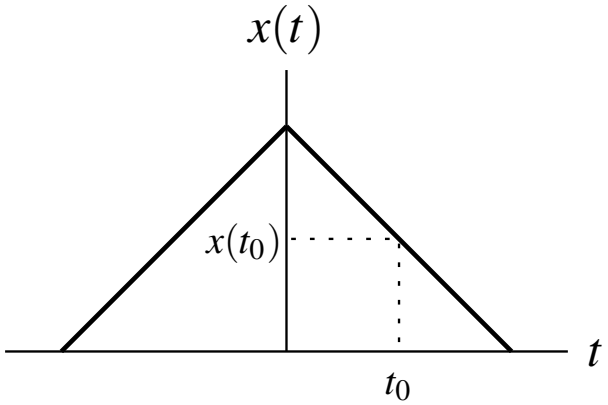
- **Scaling property.** For any nonzero real constant a,

$$\delta(at) = \frac{1}{|a|}\delta(t).$$

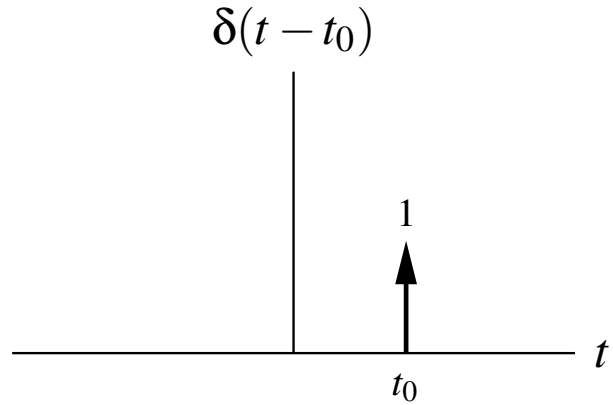
- **Even property.** The δ function is such that

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

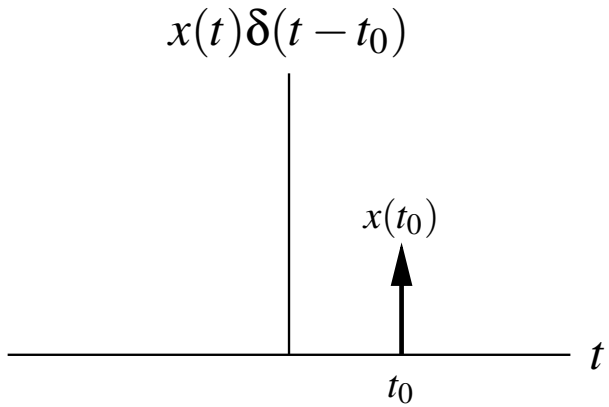
Graphical Interpretation of Equivalence Property



Function x



Time-Shifted Delta Function



Product

Representing a Rectangular Pulse (Using Unit-Step Functions)

- For real constants a and b where $a \leq b$, consider a function x of the form

$$x(t) = \begin{cases} 1 & a \leq t < b \\ 0 & \text{otherwise} \end{cases}$$

  (i.e., x is a ***rectangular pulse*** of height one, with a ***rising edge at*** a and ***falling edge at*** b).

- The function x can be equivalently written as

$$x(t) = u(t-a) - u(t-b)$$

  (i.e., the difference of two time-shifted unit-step functions).

- Unlike the original expression for x, this latter expression for x ***does not involve multiple cases***.

- In effect, by using unit-step functions, we have collapsed a formula involving multiple cases into a single expression.

Representing Functions Using Unit-Step Functions

- The idea from the previous slide can be extended to handle any function that is defined in a ***piecewise manner*** (i.e., via an expression involving multiple cases).
- That is, by using unit-step functions, we can always collapse a formula involving multiple cases into a single expression.
- Often, simplifying a formula in this way can be quite beneficial.

Section 3.4

Continuous-Time (CT) Systems

CT Systems

- A system with input x and output y can be described by the equation

$$y = \mathcal{H}x,$$

  where $\mathcal{H}$ denotes an operator (i.e., transformation).
- Note that the operator $\mathcal{H}$ ***maps a function to a function*** (not a number to a number).
- Alternatively, we can express the above relationship using the notation

$$x \xrightarrow{\mathcal{H}} y.$$

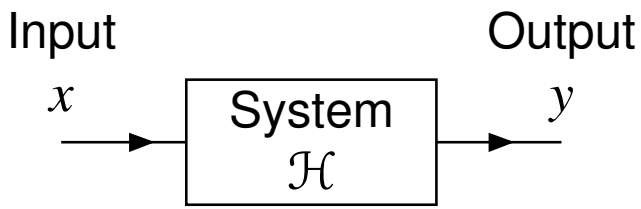
- If clear from the context, the operator $\mathcal{H}$ is often omitted, yielding the abbreviated notation

$$x \to y.$$

- Note that the symbols "$\to$" and "=" have ***very different*** meanings.
- The symbol "$\to$" should be read as "***produces***" (not as "equals").

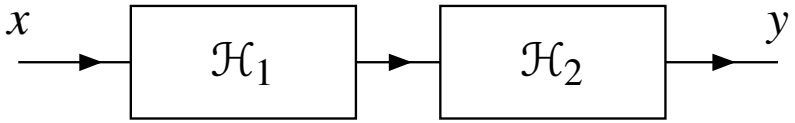
Block Diagram Representations

- Often, a system defined by the operator $\mathcal{H}$ and having the input x and output y is represented in the form of a ***block diagram*** as shown below.

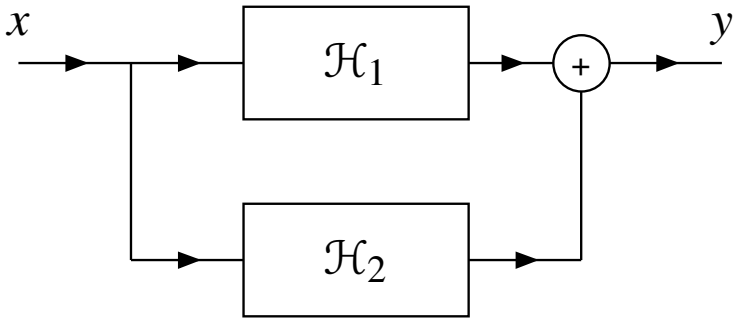


Interconnection of Systems

- ***Two basic ways*** in which systems can be interconnected are shown below.



Series



Parallel

- A **series** (or **cascade**) connection ties the output of one system to the input of the other.
- The overall series-connected system is described by the equation

$$y = \mathcal{H}_2\mathcal{H}_1 x.$$

- A **parallel** connection ties the inputs of both systems together and sums their outputs.
- The overall parallel-connected system is described by the equation

$$y = \mathcal{H}_1 x + \mathcal{H}_2 x.$$

Section 3.5

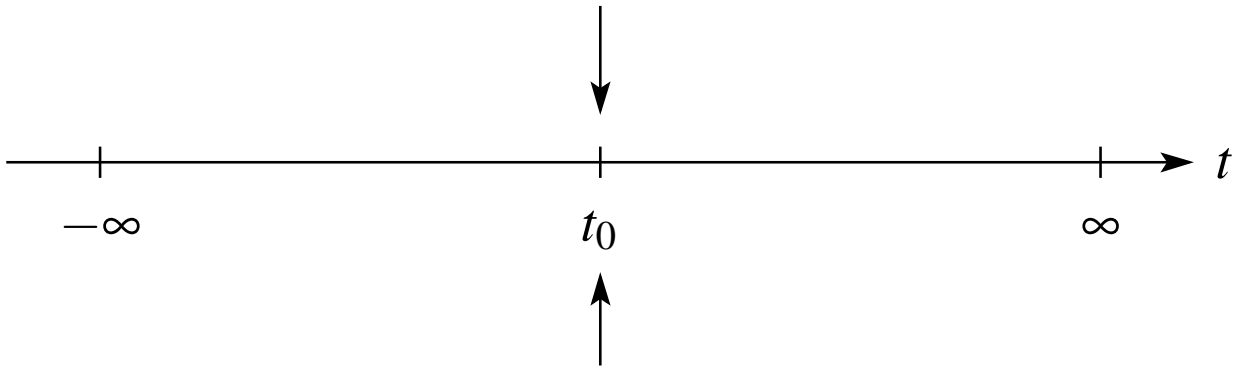
Properties of (CT) Systems

Memory

- A system $\mathcal{H}$ is said to be **memoryless** if, for every real constant t_0, $\mathcal{H}x(t_0)$ does not depend on $x(t)$ for some $t \neq t_0$.
- In other words, a memoryless system is such that the value of its output at any given point in time can depend on the value of its input at only the ***same*** point in time.
- A system that is not memoryless is said to have **memory**.
- Although simple, a memoryless system is ***not very flexible***, since its current output value cannot rely on past or future values of the input.

Memory (Continued)

If the system $\mathcal{H}$ is memoryless,
the output $\mathcal{H}x$ at t_0
can depend on the input x
only at t_0.



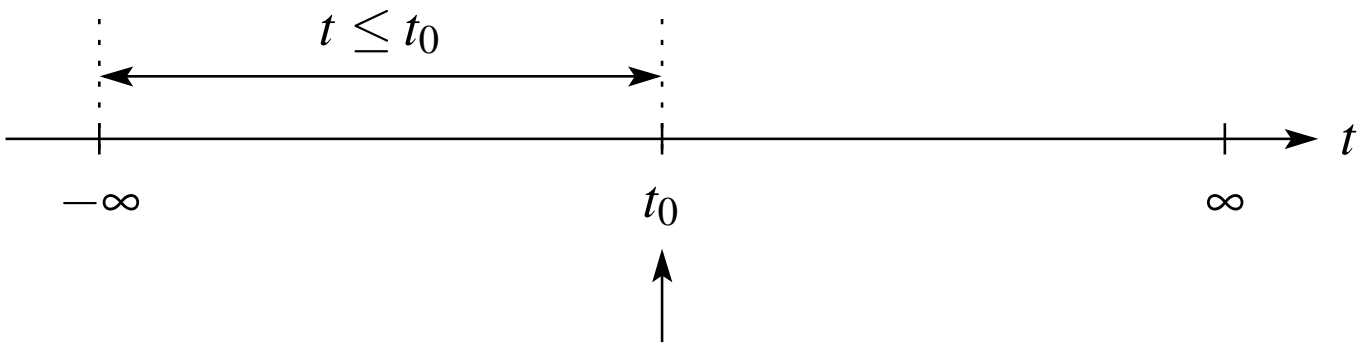
Consider the calculation of the
output $\mathcal{H}x$ at t_0.

Causality

- A system $\mathcal{H}$ is said to be **causal** if, for every real constant t_0, $\mathcal{H}x(t_0)$ does not depend on $x(t)$ for some $t > t_0$.
- In other words, a causal system is such that the value of its output at any given point in time can depend on the value of its input at only the ***same or earlier points*** in time (i.e., ***not later points in time***).
- If the independent variable t represents time, a system must be causal in order to be ***physically realizable***.
- Noncausal systems can sometimes be useful in practice, however, since the independent variable ***need not always represent time*** (e.g., the independent variable might represent position).
- A memoryless system is always causal, although the converse is not necessarily true.

Causality (Continued)

If the system $\mathcal{H}$ is causal,
the output $\mathcal{H}x$ at t_0
can depend on the input x
only at points $t \leq t_0$.



Consider the calculation of the
output $\mathcal{H}x$ at t_0.

Invertibility

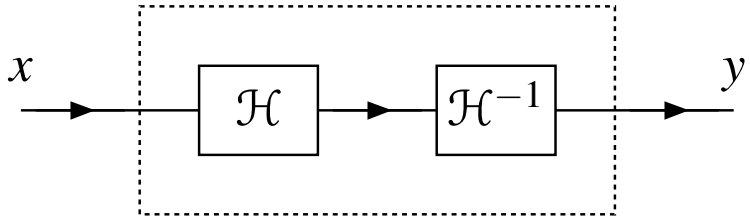
- The **inverse** of a system $\mathcal{H}$ (if it exists) is another system $\mathcal{H}^{-1}$ such that, for every function x,

$$\mathcal{H}^{-1}\mathcal{H}x = x$$

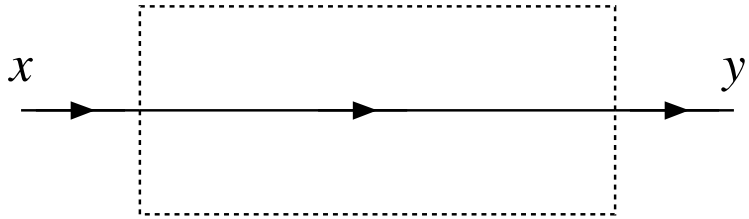
  (i.e., the system formed by the cascade interconnection of $\mathcal{H}$ followed by $\mathcal{H}^{-1}$ is a system whose input and output are equal).
- A system is said to be **invertible** if it has a corresponding inverse system (i.e., its inverse exists).
- Equivalently, a system is invertible if its input can always be ***uniquely*** determined from its output.
- An invertible system will always produce ***distinct outputs*** from any two ***distinct inputs*** (i.e., $x_1 \neq x_2 \Rightarrow \mathcal{H}x_1 \neq \mathcal{H}x_2$).
- To show that a system is ***invertible***, we simply find the ***inverse system***.
- To show that a system is ***not invertible***, we find ***two distinct inputs*** that result in ***identical outputs*** (i.e., $x_1 \neq x_2$ and $\mathcal{H}x_1 = \mathcal{H}x_2$).
- In practical terms, invertible systems are "nice" in the sense that their ***effects can be undone***.

Invertibility (Continued)

- A system $\mathcal{H}^{-1}$ being the inverse of $\mathcal{H}$ means that the following two systems are equivalent (i.e., $\mathcal{H}^{-1}\mathcal{H}$ ***is an identity***):



System 1: $y = \mathcal{H}^{-1}\mathcal{H}x$



System 2: $y = x$

Bounded-Input Bounded-Output (BIBO) Stability

- A system $\mathcal{H}$ is said to be **bounded-input bounded-output (BIBO) stable** if, for every bounded function x, $\mathcal{H}x$ is bounded (i.e., $|x(t)| < \infty$ for all t implies that $|\mathcal{H}x(t)| < \infty$ for all t).
- In other words, a BIBO stable system is such that it guarantees to always produce a bounded output as long as its input is bounded.
- To show that a system is ***BIBO stable***, we must show that ***every bounded input*** leads to a ***bounded output***.
- To show that a system is ***not BIBO stable***, we only need to find a single ***bounded input*** that leads to an ***unbounded output***.
- In practical terms, a BIBO stable system is ***well behaved*** in the sense that, as long as the system input is finite everywhere (in its domain), the output will also be finite everywhere.
- Usually, a system that is not BIBO stable will have ***serious safety issues***.
- For example, a portable music player with a battery input of 3.7 volts and headset output of ∞ volts would result in one vaporized human (and likely a big lawsuit as well).

Time Invariance (TI)

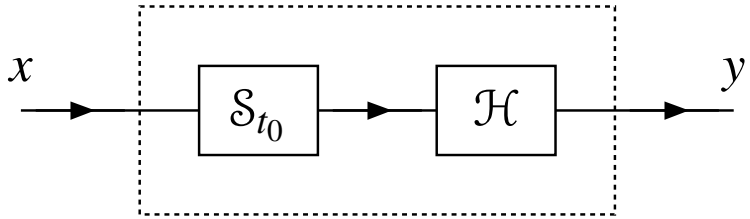
- A system $\mathcal{H}$ is said to be **time invariant (TI)** (or **shift invariant (SI)**) if, for every function x and every real constant t_0, the following condition holds:

$$\mathcal{H}x(t-t_0) = \mathcal{H}x'(t) \text{ for all } t, \quad \text{where} \quad x'(t) = x(t-t_0)$$

  (i.e., $\mathcal{H}$ ***commutes with time shifts***).
- In other words, a system is time invariant if a time shift (i.e., advance or delay) in the input always results only in an ***identical time shift*** in the output.
- A system that is not time invariant is said to be **time varying**.
- In simple terms, a time invariant system is a system whose behavior ***does not change*** with respect to time.
- Practically speaking, compared to time-varying systems, time-invariant systems are much ***easier to design and analyze***, since their behavior does not change with respect to time.

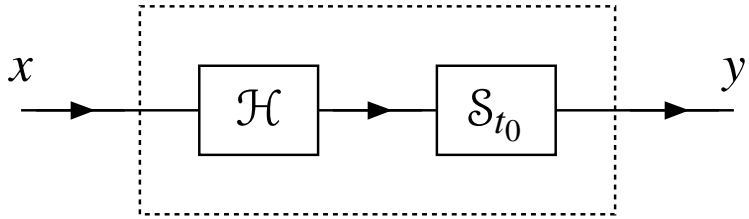
Time Invariance (Continued)

- Let $\mathcal{S}_{t_0}$ denote an operator that applies a ***time shift of*** t_0 to a function (i.e., $\mathcal{S}_{t_0}x(t) = x(t-t_0)$).
- A system $\mathcal{H}$ is ***time invariant*** if and only if the following two systems are equivalent (i.e., $\mathcal{H}$ ***commutes with*** $\mathcal{S}_{t_0}$):



System 1: $y = \mathcal{H}\mathcal{S}_{t_0}x$

$$\begin{bmatrix} y(t) = \mathcal{H}x'(t) \\ x'(t) = \mathcal{S}_{t_0}x(t) = x(t-t_0) \end{bmatrix}$$



System 2: $y = \mathcal{S}_{t_0}\mathcal{H}x$

$$\left[y(t) = \mathcal{H}x(t-t_0)\right]$$

Additivity, Homogeneity, and Linearity

- A system $\mathcal{H}$ is said to be **additive** if, for all functions x_1 and x_2, the following condition holds:

$$\mathcal{H}(x_1 + x_2) = \mathcal{H}x_1 + \mathcal{H}x_2$$

(i.e., $\mathcal{H}$ ***commutes with addition***).

- A system $\mathcal{H}$ is said to be **homogeneous** if, for every function x and every complex constant a, the following condition holds:

$$\mathcal{H}(ax) = a\mathcal{H}x$$

(i.e., $\mathcal{H}$ ***commutes with scalar multiplication***).

- A system that is both additive and homogeneous is said to be **linear**.
- In other words, a system $\mathcal{H}$ is ***linear***, if for all functions x_1 and x_2 and all complex constants a_1 and a_2, the following condition holds:

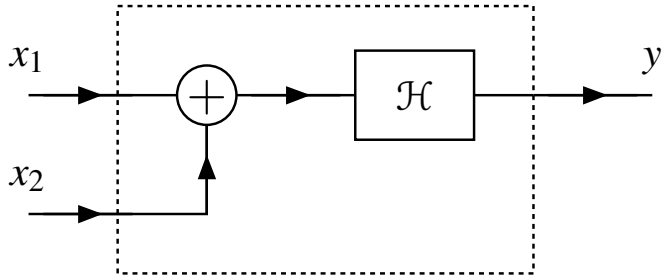
$$\mathcal{H}(a_1x_1 + a_2x_2) = a_1\mathcal{H}x_1 + a_2\mathcal{H}x_2$$

(i.e., $\mathcal{H}$ ***commutes with linear combinations***).

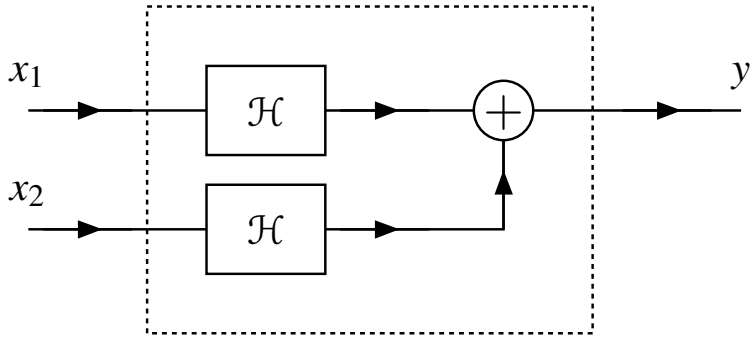
- The linearity property is also referred to as the **superposition** property.
- Practically speaking, linear systems are much ***easier to design and analyze*** than nonlinear systems.

Additivity, Homogeneity, and Linearity (Continued 1)

- The system $\mathcal{H}$ is ***additive*** if and only if the following two systems are equivalent (i.e., ***$\mathcal{H}$ commutes with addition***):

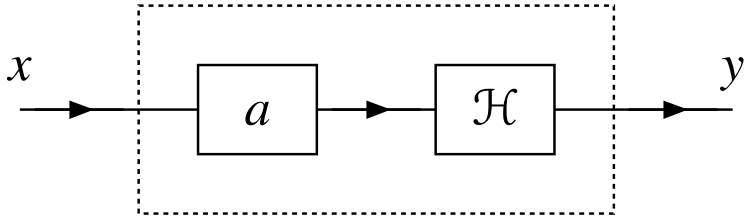


System 1: $y = \mathcal{H}(x_1 + x_2)$

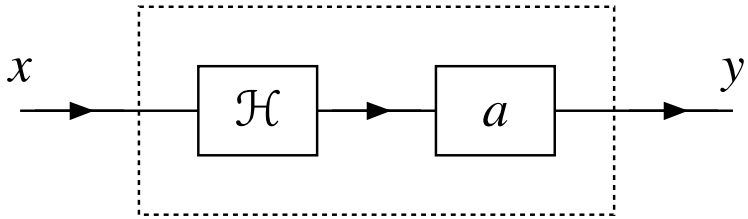


System 2: $y = \mathcal{H}x_1 + \mathcal{H}x_2$

- The system $\mathcal{H}$ is ***homogeneous*** if and only if the following two systems are equivalent (i.e., ***$\mathcal{H}$ commutes with scalar multiplication***):



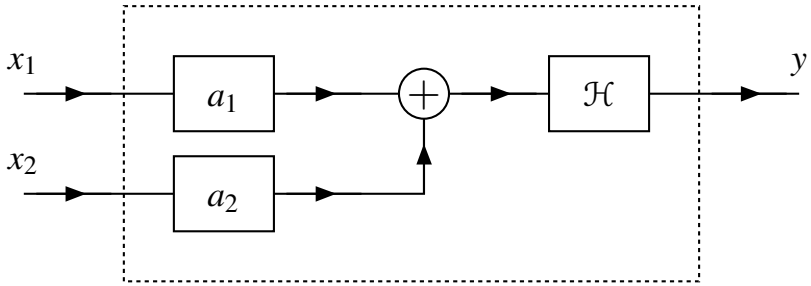
System 1: $y = \mathcal{H}(ax)$



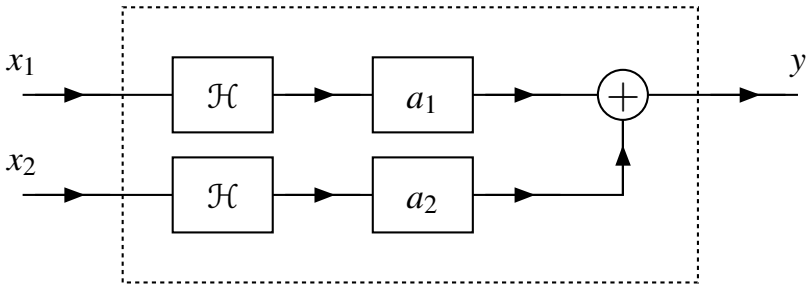
System 2: $y = a\mathcal{H}x$

Additivity, Homogeneity, and Linearity (Continued 2)

- The system $\mathcal{H}$ is ***linear*** if and only if the following two systems are equivalent (i.e., ***$\mathcal{H}$ commutes with linear combinations***):



System 1: $y = \mathcal{H}(a_1 x_1 + a_2 x_2)$



System 2: $y = a_1 \mathcal{H} x_1 + a_2 \mathcal{H} x_2$

Eigenfunctions of Systems

- A function x is said to be an **eigenfunction** of the system $\mathcal{H}$ with the **eigenvalue** λ if

$$\mathcal{H}x = \lambda x,$$

  where λ is a complex constant.
- In other words, the system $\mathcal{H}$ acts as an ideal amplifier for each of its eigenfunctions x, where the amplifier gain is given by the corresponding eigenvalue λ.
- Different systems have different eigenfunctions.
- Many of the mathematical tools developed for the study of CT systems have eigenfunctions as their basis.

Part 4

Continuous-Time Linear Time-Invariant (LTI) Systems

Why Linear Time-Invariant (LTI) Systems?

- In engineering, linear time-invariant (LTI) systems play a very important role.
- Very powerful mathematical tools have been developed for analyzing LTI systems.
- LTI systems are much easier to analyze than systems that are not LTI.
- In practice, systems that are not LTI can be well approximated using LTI models.
- So, even when dealing with systems that are not LTI, LTI systems still play an important role.

Section 4.1

Convolution

CT Convolution

- The (CT) **convolution** of the functions x and h, denoted $x * h$, is defined as the function

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

- The convolution result $x * h$ evaluated at the point t is simply a weighted average of the function x, where the weighting is given by h time reversed and shifted by t.
- Herein, the asterisk symbol (i.e., "$*$") will always be used to denote convolution, not multiplication.
- As we shall see, convolution is used extensively in systems theory.
- In particular, convolution has a special significance in the context of LTI systems.

Practical Convolution Computation

- To compute the convolution

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

we proceed as follows:

1. Plot $x(\tau)$ and $h(t-\tau)$ as a function of τ.
2. Initially, consider an arbitrarily large negative value for t. This will result in $h(t-\tau)$ being shifted very far to the left on the time axis.
3. Write the mathematical expression for $x * h(t)$.
4. Increase t gradually until the expression for $x * h(t)$ changes form. Record the interval over which the expression for $x * h(t)$ was valid.
5. Repeat steps 3 and 4 until t is an arbitrarily large positive value. This corresponds to $h(t-\tau)$ being shifted very far to the right on the time axis.
6. The results for the various intervals can be combined in order to obtain an expression for $x * h(t)$ for all t.

Properties of Convolution

- The convolution operation is ***commutative***. That is, for any two functions x and h,

$$x * h = h * x.$$

- The convolution operation is ***associative***. That is, for any functions x, h_1, and h_2,

$$(x * h_1) * h_2 = x * (h_1 * h_2).$$

- The convolution operation is ***distributive*** with respect to addition. That is, for any functions x, h_1, and h_2,

$$x * (h_1 + h_2) = x * h_1 + x * h_2.$$

Representation of Functions Using Impulses

- For any function x,

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

- Thus, any function x can be written in terms of an expression involving δ.
- Moreover, δ is the ***convolutional identity***. That is, for any function x,

$$x * \delta = x.$$

Periodic Convolution

- The convolution of two periodic functions is usually not well defined.
- This motivates an alternative notion of convolution for periodic functions known as periodic convolution.
- The **periodic convolution** of the T-periodic functions x and h, denoted $x \circledast h$, is defined as

$$x \circledast h(t) = \int_T x(\tau) h(t-\tau) d\tau,$$

where $\int_T$ denotes integration over an interval of length T.

- The periodic convolution and (linear) convolution of the T-periodic functions x and h are related as follows:

$$x \circledast h(t) = x_0 * h(t) \quad \text{where} \quad x(t) = \sum_{k=-\infty}^{\infty} x_0(t - kT)$$

(i.e., $x_0(t)$ equals $x(t)$ over a single period of x and is zero elsewhere).

Section 4.2

Convolution and LTI Systems

Impulse Response

- The response h of a system $\mathcal{H}$ to the input δ is called the **impulse response** of the system (i.e., $h = \mathcal{H}\delta$).
- For any LTI system with input x, output y, and impulse response h, the following relationship holds:

$$y = x * h.$$

- In other words, a LTI system simply ***computes a convolution***.
- Furthermore, a LTI system is ***completely characterized*** by its impulse response.
- That is, if the impulse response of a LTI system is known, we can determine the response of the system to any input.
- Since the impulse response of a LTI system is an extremely useful quantity, we often want to determine this quantity in a practical setting.
- Unfortunately, in practice, the impulse response of a system cannot be determined directly from the definition of the impulse response.

Step Response

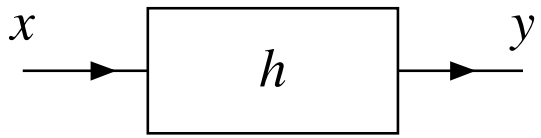
- The response s of a system $\mathcal{H}$ to the input u is called the **step response** of the system (i.e., $s = \mathcal{H}u$).
- The impulse response h and step response s of a LTI system are related as

$$h(t) = \frac{ds(t)}{dt}.$$

- Therefore, the impulse response of a system can be determined from its step response by differentiation.
- The step response provides a practical means for determining the impulse response of a system.

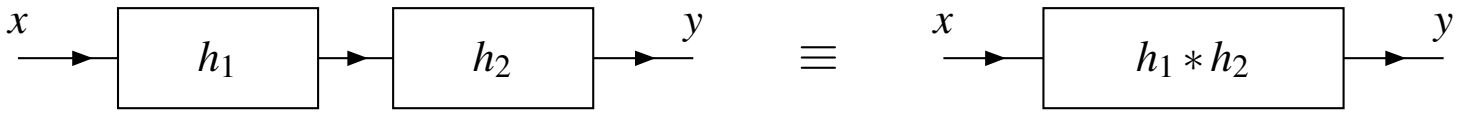
Block Diagram Representation of LTI Systems

- Often, it is convenient to represent a (CT) LTI system in block diagram form.
- Since such systems are completely characterized by their impulse response, we often label a system with its impulse response.
- That is, we represent a system with input x, output y, and impulse response h, as shown below.

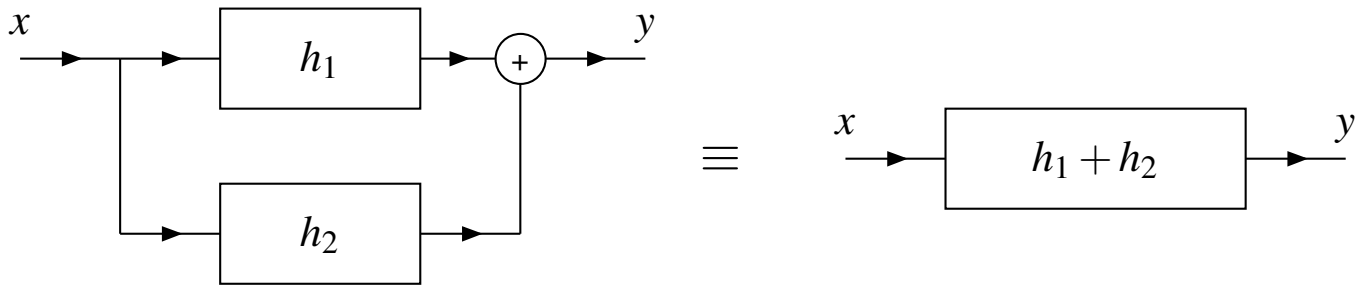


Interconnection of LTI Systems

- The ***series*** interconnection of the LTI systems with impulse responses h_1 and h_2 is the LTI system with impulse response $h_1 * h_2$. That is, we have the equivalence shown below.



- The ***parallel*** interconnection of the LTI systems with impulse responses h_1 and h_2 is the LTI system with impulse response $h_1 + h_2$. That is, we have the equivalence shown below.



Section 4.3

Properties of LTI Systems

Memory

- A LTI system with impulse response h is memoryless if and only if

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

- That is, a LTI system is memoryless if and only if its impulse response h is of the form

$$h(t) = K\delta(t),$$

  where K is a complex constant.
- Consequently, every memoryless LTI system with input x and output y is characterized by an equation of the form

$$y = x * (K\delta) = Kx$$

  (i.e., the system is an ideal amplifier).
- For a LTI system, the memoryless constraint is extremely restrictive (as every memoryless LTI system is an ideal amplifier).

Causality

- A LTI system with impulse response h is causal if and only if

$$h(t) = 0 \quad \text{for all } t < 0$$

(i.e., h is a causal function).

- It is due to the above relationship that we call a function x, satisfying

$$x(t) = 0 \quad \text{for all } t < 0,$$

a causal function.

Invertibility

- The inverse of a LTI system, if such a system exists, is a LTI system.
- Let h and h_{inv} denote the impulse responses of a LTI system and its (LTI) inverse, respectively. Then,

$$h * h_{\text{inv}} = \delta.$$

- Consequently, a LTI system with impulse response h is invertible if and only if there exists a function h_{inv} such that

$$h * h_{\text{inv}} = \delta.$$

- Except in simple cases, the above condition is often quite difficult to test.

BIBO Stability

- A LTI system with impulse response h is BIBO stable if and only if

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

(i.e., h is ***absolutely integrable***).

Eigenfunctions of LTI Systems

- As it turns out, every complex exponential is an eigenfunction of all LTI systems.
- For a LTI system $\mathcal{H}$ with impulse response h,

$$\mathcal{H}\{e^{st}\}(t) = H(s)e^{st},$$

where s is a complex constant and

$$H(s) = \int_{-\infty}^{\infty} h(t)e^{-st}dt.$$

- That is, e^{st} is an eigenfunction of a LTI system and $H(s)$ is the corresponding eigenvalue.
- We refer to H as the **system function** (or **transfer function**) of the system $\mathcal{H}$.
- From above, we can see that the response of a LTI system to a complex exponential is the same complex exponential multiplied by the complex factor $H(s)$.

Representations of Functions Using Eigenfunctions

- Consider a LTI system with input x, output y, and system function H.
- Suppose that the input x can be expressed as the linear combination of complex exponentials

$$x(t) = \sum_k a_k e^{s_k t},$$

where the a_k and s_k are complex constants.

- Using the fact that complex exponentials are eigenfunctions of LTI systems, we can conclude

$$y(t) = \sum_k a_k H(s_k) e^{s_k t}.$$

- Thus, if an input to a LTI system can be expressed as a linear combination of complex exponentials, the output can also be expressed as a linear combination of the ***same*** complex exponentials.
- The above formula can be used to determine the output of a LTI system from its input in a way that does not require convolution.

Part 5

Continuous-Time Fourier Series (CTFS)

Introduction

- The (CT) Fourier series is a representation for ***periodic*** functions.
- With a Fourier series, a function is represented as a ***linear combination of complex sinusoids***.
- The use of complex sinusoids is desirable due to their numerous attractive properties.
- For example, complex sinusoids are continuous and differentiable. They are also easy to integrate and differentiate.
- Perhaps, most importantly, complex sinusoids are ***eigenfunctions*** of LTI systems.

Section 5.1

Fourier Series

Harmonically-Related Complex Sinusoids

- A set of complex sinusoids is said to be **harmonically related** if there exists some constant ω_0 such that the fundamental frequency of each complex sinusoid is an integer multiple of ω_0.
- Consider the set of harmonically-related complex sinusoids given by

$$\phi_k(t) = e^{jk\omega_0 t} \quad \text{for all integer } k.$$

- The fundamental frequency of the kth complex sinusoid ϕ_k is $k\omega_0$, an integer multiple of ω_0.
- Since the fundamental frequency of each of the harmonically-related complex sinusoids is an integer multiple of ω_0, a linear combination of these complex sinusoids must be periodic.
- More specifically, a linear combination of these complex sinusoids is periodic with period $T = \frac{2\pi}{\omega_0}$.

CT Fourier Series

- A periodic (complex-valued) function x with fundamental period T and fundamental frequency $\omega_0 = \frac{2\pi}{T}$ can be represented as a linear combination of harmonically-related complex sinusoids as

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}.$$

- Such a representation is known as (the complex exponential form of) a (CT) **Fourier series**, and the c_k are called **Fourier series coefficients**.
- The above formula for x is often referred to as the **Fourier series synthesis equation**.
- The terms in the summation for $k = K$ and $k = -K$ are called the Kth **harmonic components**, and have the fundamental frequency $K\omega_0$.
- To denote that a function x has the Fourier series coefficient sequence c_k, we write

$$x(t) \overset{\text{CTFS}}{\longleftrightarrow} c_k.$$

CT Fourier Series (Continued)

- The periodic function x with fundamental period T and fundamental frequency $\omega_0 = \frac{2\pi}{T}$ has the Fourier series coefficients c_k given by

$$c_k = \frac{1}{T} \int_T x(t) e^{-jk\omega_0 t} dt,$$

  where $\int_T$ denotes integration over an arbitrary interval of length T (i.e., one period of x).
- The above equation for c_k is often referred to as the **Fourier series analysis equation**.

Section 5.2

Convergence Properties of Fourier Series

Remarks on Equality of Functions

- The equality of functions can be defined in more than one way.
- Two functions x and y are said to be **equal in the pointwise sense** if $x(t) = y(t)$ for all t (i.e., x and y are equal at every point).
- Two functions x and y are said to be **equal in the mean-squared error (MSE) sense** if $\int |x(t) - y(t)|^2 \, dt = 0$ (i.e., the energy in $x - y$ is zero).
- Pointwise equality is a stronger condition than MSE equality (i.e., pointwise equality implies MSE equality but the converse is not true).
- Consider the functions

$$x_1(t) = 1 \text{ for all } t, \quad x_2(t) = 1 \text{ for all } t, \quad \text{and}$$

$$x_3(t) = \begin{cases} 2 & t = 0 \\ 1 & \text{otherwise.} \end{cases}$$

- The functions x_1 and x_2 are equal in both the pointwise sense and MSE sense.
- The functions x_1 and x_3 are equal in the MSE sense, but not in the pointwise sense.

Convergence of Fourier Series

- Since a Fourier series can have an infinite number of (nonzero) terms, and an infinite sum may or may not converge, we need to consider the issue of convergence.
- That is, when we claim that a periodic function x is equal to the Fourier series $\sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}$, is this claim actually correct?
- Consider a periodic function x that we wish to represent with the Fourier series

$$\sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}.$$

- Let x_N denote the Fourier series truncated after the Nth harmonic components as given by

$$x_N(t) = \sum_{k=-N}^{N} c_k e^{jk\omega_0 t}.$$

- Here, we are interested in whether $\lim_{N\to\infty} x_N$ is equal (in some sense) to x.

Convergence of Fourier Series (Continued)

- Again, let x_N denote the Fourier series for the periodic function x truncated after the Nth harmonic components as given by

$$x_N(t) = \sum_{k=-N}^{N} c_k e^{jk\omega_0 t}.$$

- If $\lim_{N\to\infty} x_N(t) = x(t)$ for all t (i.e., $\lim_{N\to\infty} x_N$ is equal to x in the pointwise sense), the Fourier series is said to converge **pointwise** to x.
- If convergence is pointwise and the rate of convergence is the same everywhere, the convergence is said to be **uniform**.
- If $\lim_{N\to\infty} \frac{1}{T}\int_T |x_N(t) - x(t)|^2 dt = 0$ (i.e., $\lim_{N\to\infty} x_N$ is equal to x in the MSE sense), the Fourier series is said to converge to x in the **MSE** sense.
- Pointwise convergence is a stronger condition than MSE convergence (i.e., pointwise convergence implies MSE convergence, but the converse is not true).

Convergence of Fourier Series: Continuous Case

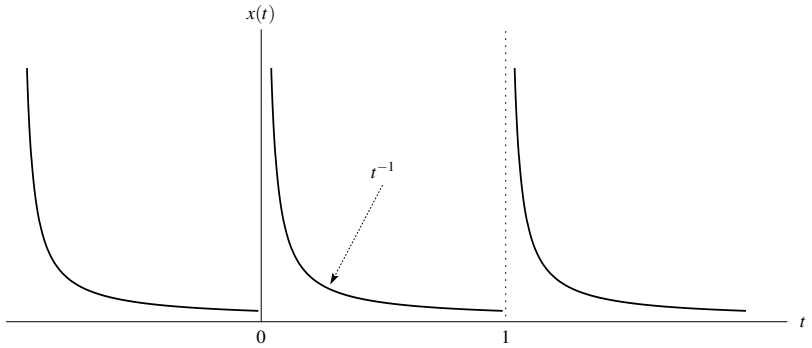
- If a periodic function x is ***continuous*** and its Fourier series coefficients c_k are ***absolutely summable*** (i.e., $\sum_{k=-\infty}^{\infty} |c_k| < \infty$), then the Fourier series representation of x converges ***uniformly*** (i.e., pointwise at the same rate everywhere).
- Since, in practice, we often encounter functions with discontinuities (e.g., a square wave), the above result is of somewhat limited value.

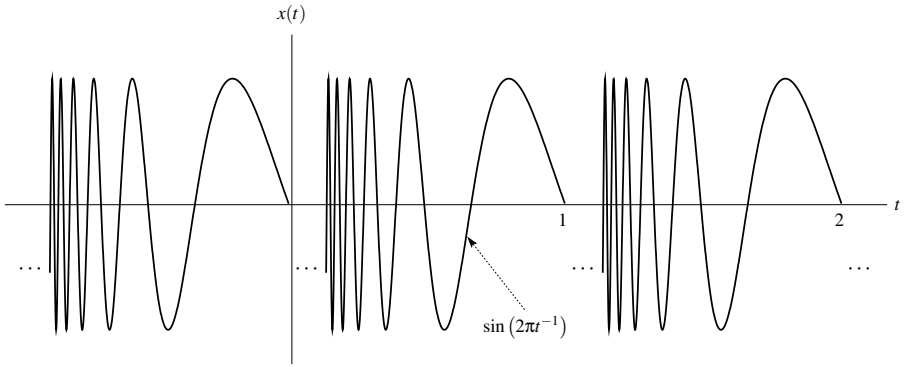
Convergence of Fourier Series: Finite-Energy Case

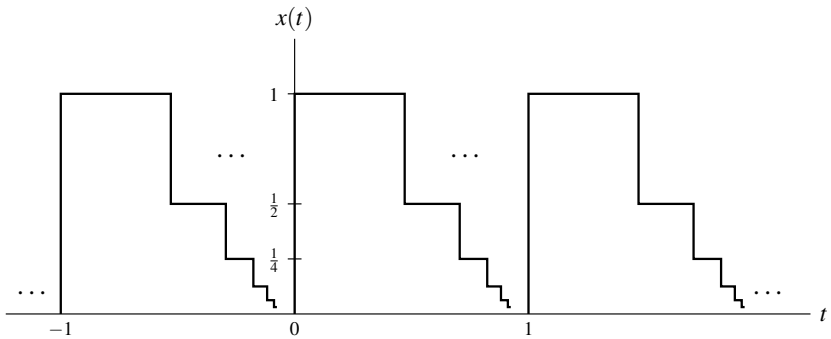
- If a periodic function x has ***finite energy*** in a single period (i.e., $\int_T |x(t)|^2 \, dt < \infty$), the Fourier series converges in the ***MSE*** sense.
- Since, in situations of practice interest, the finite-energy condition in the above theorem is typically satisfied, the theorem is usually applicable.
- It is important to note, however, that MSE convergence (i.e., $E = 0$) does not necessarily imply pointwise convergence (i.e., $\tilde{x}(t) = x(t)$ for all t).
- Thus, the above convergence theorem does not provide much useful information regarding the value of $\tilde{x}(t)$ for specific values of t.
- Consequently, the above theorem is typically most useful for simply determining if the Fourier series converges.

Dirichlet Conditions

- The **Dirichlet conditions** for the periodic function x are as follows:
  1. over a single period, x is ***absolutely integrable*** (i.e., $\int_T |x(t)|\,dt < \infty$);
  2. over a single period, x has a finite number of maxima and minima (i.e., x is of ***bounded variation***); and
  3. over any finite interval, x has a ***finite number of discontinuities***, each of which is ***finite***.
- Examples of functions violating the Dirichlet conditions are shown below.







Convergence of Fourier Series: Dirichlet Case

- If a periodic function x satisfies the ***Dirichlet conditions***, then:
  1. the Fourier series converges pointwise everywhere to x, except at the points of discontinuity of x; and
  2. at each point t_a of discontinuity of x, the Fourier series $\tilde{x}$ converges to

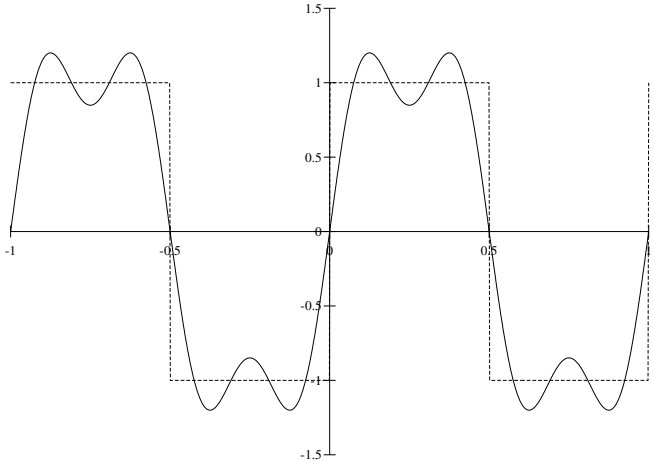
$$\tilde{x}(t_a) = \frac{1}{2}\left[x(t_a^-) + x(t_a^+)\right],$$

     where $x(t_a^-)$ and $x(t_a^+)$ denote the values of the function x on the left- and right-hand sides of the discontinuity, respectively.
- Since most functions tend to satisfy the Dirichlet conditions and the above convergence result specifies the value of the Fourier series at every point, this result is often very useful in practice.

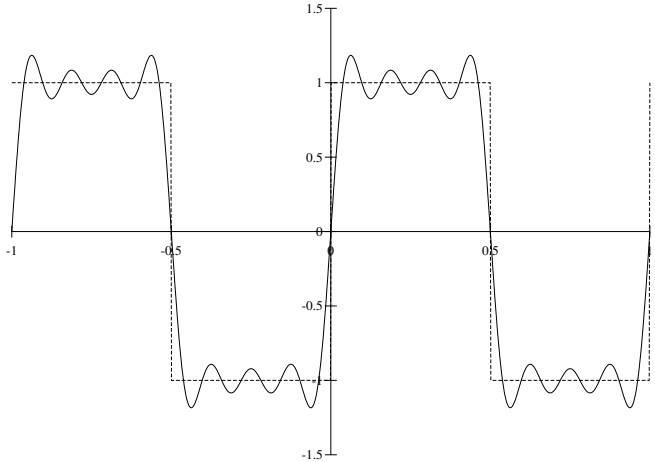
Gibbs Phenomenon

- In practice, we frequently encounter functions with discontinuities.
- When a function x has discontinuities, the Fourier series representation of x does not converge uniformly (i.e., at the same rate everywhere).
- The rate of convergence is much slower at points in the vicinity of a discontinuity.
- Furthermore, in the vicinity of a discontinuity, the truncated Fourier series x_N exhibits ripples, where the peak amplitude of the ripples does not seem to decrease with increasing N.
- As it turns out, as N increases, the ripples get compressed towards discontinuity, but, for any finite N, the peak amplitude of the ripples remains approximately constant.
- This behavior is known as **Gibbs phenomenon**.
- The above behavior is one of the weaknesses of Fourier series (i.e., Fourier series converge very slowly near discontinuities).

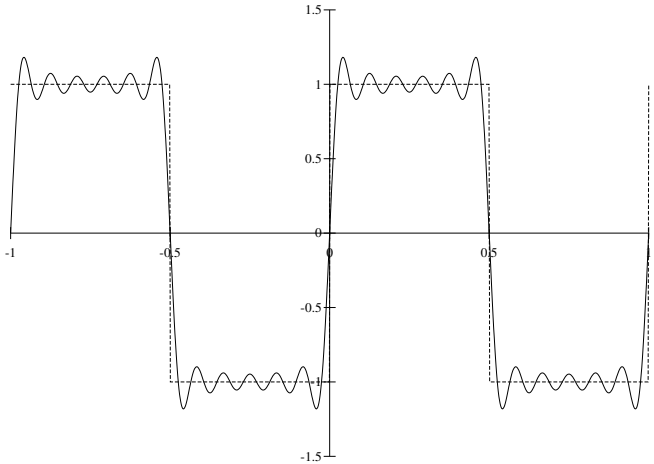
Gibbs Phenomenon: Periodic Square Wave Example



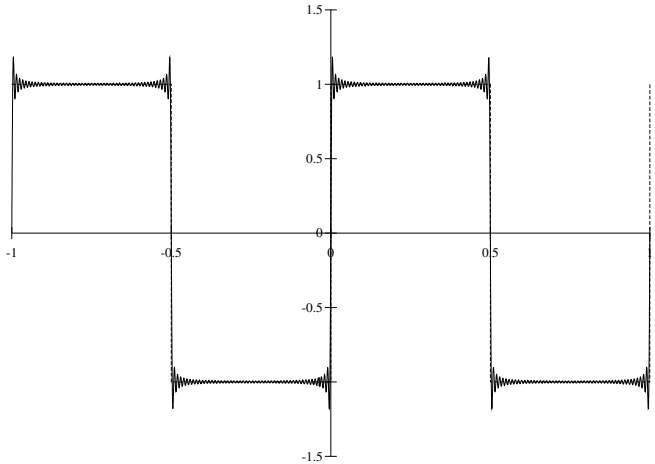
Fourier series truncated after the 3rd harmonic components



Fourier series truncated after the 7th harmonic components



Fourier series truncated after the 11th harmonic components



Fourier series truncated after the 101st harmonic components

Section 5.3

Properties of Fourier Series

Properties of (CT) Fourier Series

$$x(t) \overset{\text{CTFS}}{\longleftrightarrow} a_k \quad \text{and} \quad y(t) \overset{\text{CTFS}}{\longleftrightarrow} b_k$$

Property	Time Domain	Fourier Domain
Linearity	$\alpha x(t) + \beta y(t)$	$\alpha a_k + \beta b_k$
Translation	$x(t - t_0)$	$e^{-jk(2\pi/T)t_0} a_k$
Modulation	$e^{jM(2\pi/T)t} x(t)$	a_{k-M}
Reflection	$x(-t)$	a_{-k}
Conjugation	$x^*(t)$	a^*_{-k}
Periodic Convolution	$x \circledast y(t)$	$T a_k b_k$
Multiplication	$x(t)y(t)$	$\sum_{n=-\infty}^{\infty} a_n b_{k-n}$

Property	
Parseval's Relation	$\frac{1}{T}\int_T \lvert x(t)\rvert^2 dt = \sum_{k=-\infty}^{\infty} \lvert a_k\rvert^2$
Even Symmetry	x is even $\Leftrightarrow$ a is even
Odd Symmetry	x is odd $\Leftrightarrow$ a is odd
Real / Conjugate Symmetry	x is real $\Leftrightarrow$ a is conjugate symmetric

Linearity

- Let x and y be two periodic functions with the same period. If $x(t) \overset{\text{CTFS}}{\longleftrightarrow} a_k$ and $y(t) \overset{\text{CTFS}}{\longleftrightarrow} b_k$, then

$$\alpha x(t) + \beta y(t) \overset{\text{CTFS}}{\longleftrightarrow} \alpha a_k + \beta b_k,$$

  where α and β are complex constants.
- That is, a linear combination of functions produces the same linear combination of their Fourier series coefficients.

Time Shifting (Translation)

- Let x denote a periodic function with period T and the corresponding frequency $\omega_0 = 2\pi/T$. If $x(t) \overset{\text{CTFS}}{\longleftrightarrow} c_k$, then

$$x(t - t_0) \overset{\text{CTFS}}{\longleftrightarrow} e^{-jk\omega_0 t_0} c_k = e^{-jk(2\pi/T)t_0} c_k,$$

  where t_0 is a real constant.
- In other words, time shifting a periodic function changes the argument (but not magnitude) of its Fourier series coefficients.

Frequency Shifting (Modulation)

- Let x denote a periodic function with period T and the corresponding frequency $\omega_0 = 2\pi/T$. If $x(t) \overset{\text{CTFS}}{\longleftrightarrow} c_k$, then

$$e^{jM(2\pi/T)t}x(t) = e^{jM\omega_0 t}x(t) \overset{\text{CTFS}}{\longleftrightarrow} c_{k-M},$$

  where M is an integer constant.
- In other words, multiplying a periodic function by $e^{jM\omega_0 t}$ shifts the Fourier-series coefficient sequence.

Time Reversal (Reflection)

- Let x denote a periodic function with period T and the corresponding frequency $\omega_0 = 2\pi/T$. If $x(t) \overset{\text{CTFS}}{\longleftrightarrow} c_k$, then

$$x(-t) \overset{\text{CTFS}}{\longleftrightarrow} c_{-k}.$$

- That is, time reversal of a function results in a time reversal of its Fourier series coefficients.

Conjugation

- For a T-periodic function x with Fourier series coefficient sequence c, the following property holds:

$$x^*(t) \overset{\text{CTFS}}{\longleftrightarrow} c^*_{-k}$$

- In other words, conjugating a function has the effect of time reversing and conjugating the Fourier series coefficient sequence.

Periodic Convolution

- Let x and y be two periodic functions with the same period T. If $x(t) \overset{\text{CTFS}}{\longleftrightarrow} a_k$ and $y(t) \overset{\text{CTFS}}{\longleftrightarrow} b_k$, then

$$x \circledast y(t) \overset{\text{CTFS}}{\longleftrightarrow} T a_k b_k.$$

- In other words, periodic convolution of two functions corresponds to the multiplication (up to a scale factor) of their Fourier-series coefficient sequences.

Multiplication

- Let x and y be two periodic functions with the same period. If $x(t) \overset{\text{CTFS}}{\longleftrightarrow} a_k$ and $y(t) \overset{\text{CTFS}}{\longleftrightarrow} b_k$, then

$$x(t)y(t) \overset{\text{CTFS}}{\longleftrightarrow} \sum_{n=-\infty}^{\infty} a_n b_{k-n}$$

- As we shall see later, the above summation is the DT convolution of a and b.
- In other words, the multiplication of two periodic functions corresponds to the DT convolution of their corresponding Fourier-series coefficient sequences.

Parseval's Relation

- A function x and its Fourier series coefficient sequence a satisfy the following relationship:

$$\frac{1}{T}\int_T |x(t)|^2\,dt = \sum_{k=-\infty}^{\infty} |a_k|^2 .$$

- The above relationship is simply stating that the amount of energy in x (i.e., $\frac{1}{T}\int_T |x(t)|^2\,dt$) and the amount of energy in the Fourier series coefficient sequence a (i.e., $\sum_{k=-\infty}^{\infty} |a_k|^2$) are equal.
- In other words, the transformation between a function and its Fourier series coefficient sequence preserves energy.

Even and Odd Symmetry

- For a periodic function x with Fourier series coefficient sequence c, the following properties hold:

$$x \text{ is even} \Leftrightarrow c \text{ is even}; \quad \text{and}$$

$$x \text{ is odd} \Leftrightarrow c \text{ is odd}.$$

- In other words, the even/odd symmetry properties of x and c always match.

Real Functions

- A function x is ***real*** if and only if its Fourier series coefficient sequence c satisfies

$$c_k = c_{-k}^* \text{ for all } k$$

  (i.e., c is ***conjugate symmetric***).
- Thus, for a real-valued function, the negative-indexed Fourier series coefficients are ***redundant***, as they are completely determined by the nonnegative-indexed coefficients.
- From properties of complex numbers, one can show that $c_k = c_{-k}^*$ is equivalent to

$$|c_k| = |c_{-k}| \quad \text{and} \quad \arg c_k = -\arg c_{-k}$$

  (i.e., $|c_k|$ is ***even*** and $\arg c_k$ is ***odd***).
- Note that x being real does ***not*** necessarily imply that c is real.

Trigonometric Forms of a Fourier Series

- Consider the periodic function x with the Fourier series coefficients c_k.
- If x is real, then its Fourier series can be rewritten in two other forms, known as the combined trigonometric and trigonometric forms.
- The **combined trigonometric form** of a Fourier series has the appearance

$$x(t) = c_0 + 2\sum_{k=1}^{\infty} |c_k| \cos(k\omega_0 t + \theta_k),$$

  where $\theta_k = \arg c_k$.
- The **trigonometric form** of a Fourier series has the appearance

$$x(t) = c_0 + \sum_{k=1}^{\infty} \left[\alpha_k \cos(k\omega_0 t) + \beta_k \sin(k\omega_0 t)\right],$$

  where $\alpha_k = 2\operatorname{Re} c_k$ and $\beta_k = -2\operatorname{Im} c_k$.
- Note that the trigonometric forms contain only ***real*** quantities.

Other Properties of Fourier Series

- For a T-periodic function x with Fourier-series coefficient sequence c, the following properties hold:
  1. c_0 is the average value of x over a single period T;
  2. x is real and even $\Leftrightarrow$ c is real and even; and
  3. x is real and odd $\Leftrightarrow$ c is purely imaginary and odd.

Section 5.4

Fourier Series and Frequency Spectra

A New Perspective on Functions: The Frequency Domain

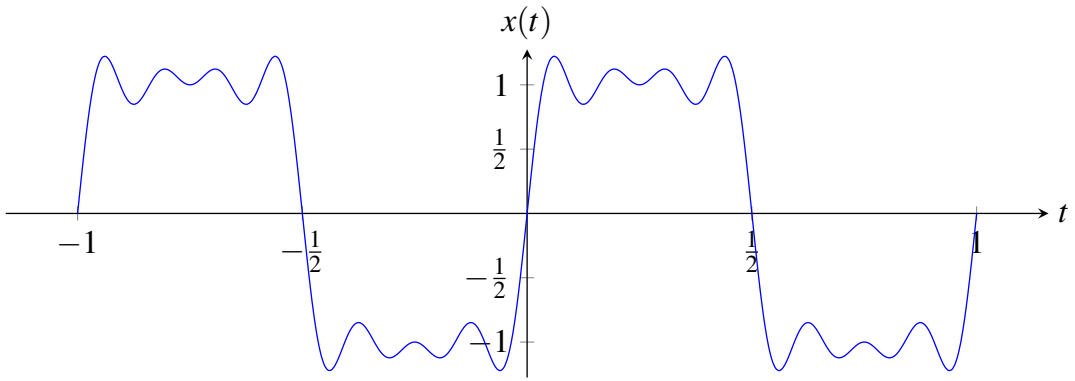
- The Fourier series provides us with an entirely new way to view functions.
- Instead of viewing a function as having information distributed with respect to ***time*** (i.e., a function whose domain is time), we view a function as having information distributed with respect to ***frequency*** (i.e., a function whose domain is frequency).
- This so called frequency-domain perspective is of fundamental importance in engineering.
- Many engineering problems can be solved ***much more easily*** using the frequency domain than the time domain.
- The Fourier series coefficients of a function x provide a means to ***quantify*** how much information x has at different frequencies.
- The distribution of information in a function over different frequencies is referred to as the ***frequency spectrum*** of the function.

Motivating Example

- Consider the real 1-periodic function x having the Fourier series representation

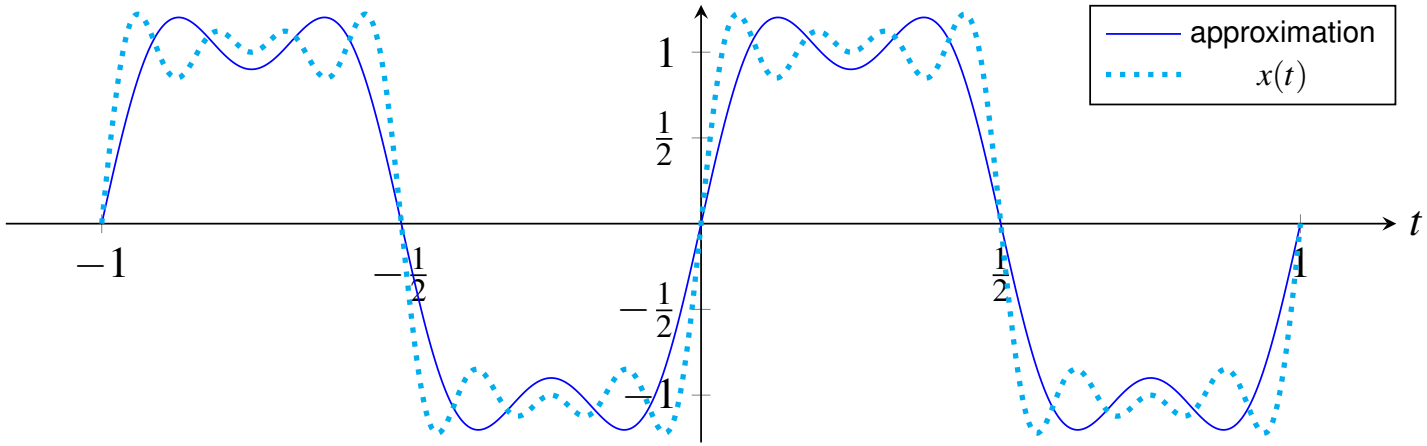
$$x(t) = -\frac{j}{10}e^{-j14\pi t} - \frac{2j}{10}e^{-j10\pi t} - \frac{4j}{10}e^{-j6\pi t} - \frac{13j}{10}e^{-j2\pi t} + \frac{13j}{10}e^{j2\pi t} + \frac{4j}{10}e^{j6\pi t} + \frac{2j}{10}e^{j10\pi t} + \frac{j}{10}e^{j14\pi t}.$$

- A plot of x is shown below.

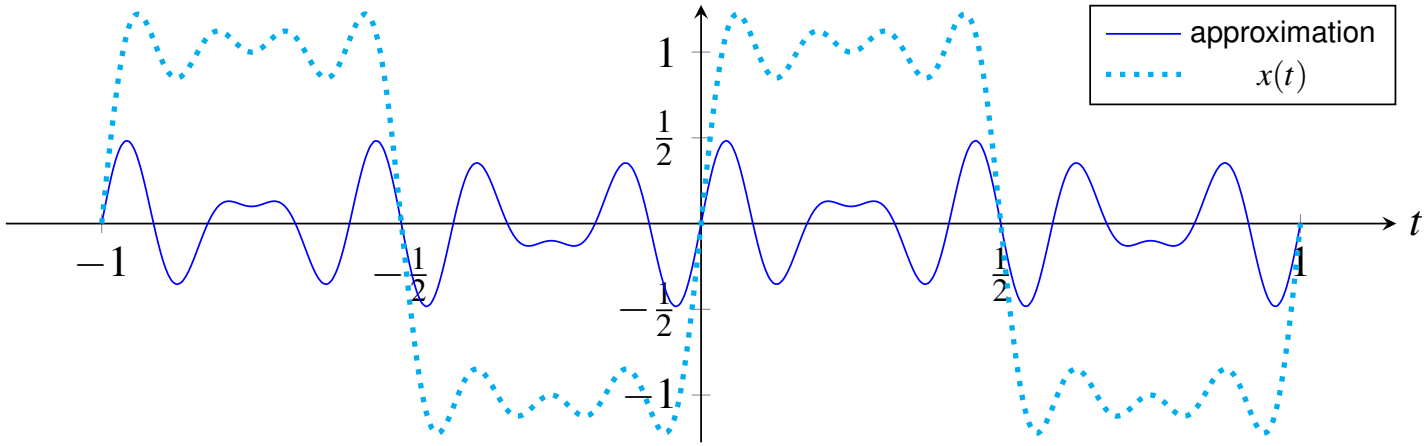


- The terms that make the most dominant contribution to the overall sum are the ones with the largest magnitude coefficients.
- To illustrate this, we consider the problem of determining the best approximation of x that keeps only 4 of the 8 terms in the Fourier series.

Motivating Example (Continued)



Approximation using the 4 terms with the largest magnitude coefficients



Approximation using the 4 terms with the smallest magnitude nonzero coefficients

- To gain further insight into the role played by the Fourier series coefficients c_k in the context of the frequency spectrum of the function x, it is helpful to write the Fourier series with the c_k expressed in ***polar form*** as follows:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} = \sum_{k=-\infty}^{\infty} |c_k| \, e^{j(k\omega_0 t + \arg c_k)}.$$

- Clearly, the kth term in the summation corresponds to a complex sinusoid with fundamental frequency $k\omega_0$ that has been ***amplitude scaled*** by a factor of $|c_k|$ and ***time shifted*** by an amount that depends on $\arg c_k$.
- For a given k, the ***larger*** $|c_k|$ is, the larger is the amplitude of its corresponding complex sinusoid $e^{jk\omega_0 t}$, and therefore the ***larger the contribution*** the kth term (which is associated with frequency $k\omega_0$) will make to the overall summation.
- In this way, we can use $|c_k|$ as a ***measure*** of how much information a function x has at the frequency $k\omega_0$.

Fourier Series and Frequency Spectra (Continued)

- The Fourier series coefficients c_k are referred to as the **frequency spectrum** of x.
- The magnitudes $|c_k|$ of the Fourier series coefficients are referred to as the **magnitude spectrum** of x.
- The arguments $\arg c_k$ of the Fourier series coefficients are referred to as the **phase spectrum** of x.
- Normally, the spectrum of a function is plotted against frequency $k\omega_0$ instead of k.
- Since the Fourier series only has frequency components at integer multiples of the fundamental frequency, the frequency spectrum is ***discrete*** in the independent variable (i.e., frequency).
- Due to the general appearance of frequency-spectrum plot (i.e., a number of vertical lines at various frequencies), we refer to such spectra as **line spectra**.

Frequency Spectra of Real Functions

- Recall that, for a real function x, the Fourier series coefficient sequence c satisfies

$$c_k = c_{-k}^*$$

(i.e., c is ***conjugate symmetric***), which is equivalent to

$$|c_k| = |c_{-k}| \quad \text{and} \quad \arg c_k = -\arg c_{-k}.$$

- Since $|c_k| = |c_{-k}|$, the magnitude spectrum of a real function is always ***even***.
- Similarly, since $\arg c_k = -\arg c_{-k}$, the phase spectrum of a real function is always ***odd***.
- Due to the symmetry in the frequency spectra of real functions, we typically ***ignore negative frequencies*** when dealing with such functions.
- In the case of functions that are complex but not real, frequency spectra do not possess the above symmetry, and ***negative frequencies become important***.

Section 5.5

Fourier Series and LTI Systems

Frequency Response

- Recall that a LTI system $\mathcal{H}$ with impulse response h is such that $\mathcal{H}\{e^{st}\}(t) = H_{\mathrm{L}}(s)e^{st}$, where $H_{\mathrm{L}}(s) = \int_{-\infty}^{\infty} h(t)e^{-st}dt$. (That is, complex exponentials are ***eigenfunctions*** of LTI systems.)
- Since a complex sinusoid is a ***special case*** of a complex exponential, we can reuse the above result for the special case of complex sinusoids.
- For a LTI system $\mathcal{H}$ with impulse response h,

$$\mathcal{H}\{e^{j\omega t}\}(t) = H(\omega)e^{j\omega t},$$

  where ω is a real constant and

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-j\omega t}dt.$$

- That is, $e^{j\omega t}$ is an ***eigenfunction*** of a LTI system and $H(\omega)$ is the corresponding ***eigenvalue***.
- We refer to H as the **frequency response** of the system $\mathcal{H}$.

- Consider a LTI system with input x, output y, and frequency response H.
- Suppose that the T-periodic input x is expressed as the Fourier series

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}, \quad \text{where } \omega_0 = \frac{2\pi}{T}.$$

- Using our knowledge about the ***eigenfunctions*** of LTI systems, we can conclude

$$y(t) = \sum_{k=-\infty}^{\infty} c_k H(k\omega_0) e^{jk\omega_0 t}.$$

- Thus, if the input x to a LTI system is a Fourier series, the output y is also a Fourier series. More specifically, if $x(t) \overset{\text{CTFS}}{\longleftrightarrow} c_k$ then $y(t) \overset{\text{CTFS}}{\longleftrightarrow} H(k\omega_0)c_k$.
- The above formula can be used to determine the output of a LTI system from its input in a way that ***does not require convolution***.

Filtering

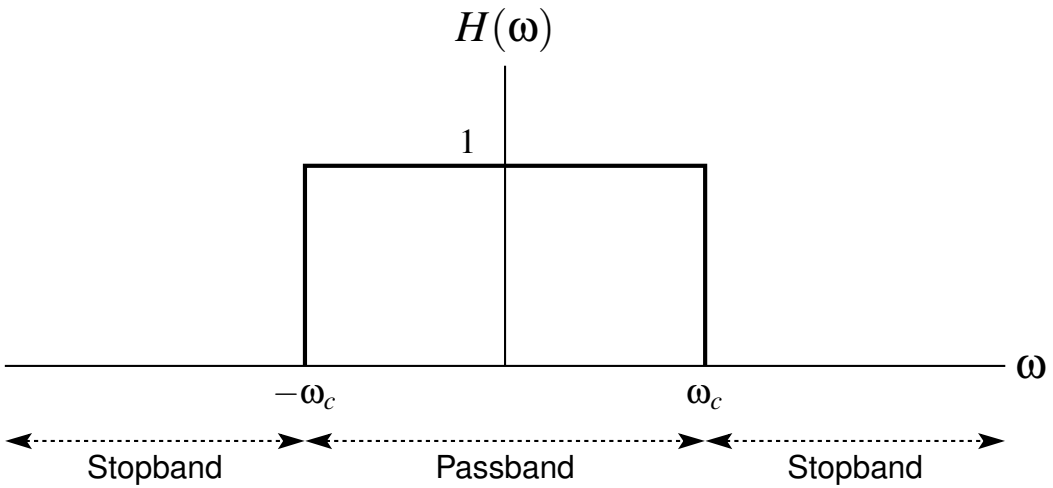
- In many applications, we want to ***modify the spectrum*** of a function by either amplifying or attenuating certain frequency components.
- This process of modifying the frequency spectrum of a function is called **filtering**.
- A system that performs a filtering operation is called a **filter**.
- Many types of filters exist.
- **Frequency selective filters** pass some frequencies with little or no distortion, while significantly attenuating other frequencies.
- Several basic types of frequency-selective filters include: lowpass, highpass, and bandpass.

Ideal Lowpass Filter

- An **ideal lowpass filter** eliminates all frequency components with a frequency whose magnitude is greater than some cutoff frequency, while leaving the remaining frequency components unaffected.
- Such a filter has a ***frequency response*** of the form

$$H(\omega) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & \text{otherwise,} \end{cases}$$

  where ω_c is the **cutoff frequency**.
- A plot of this frequency response is given below.

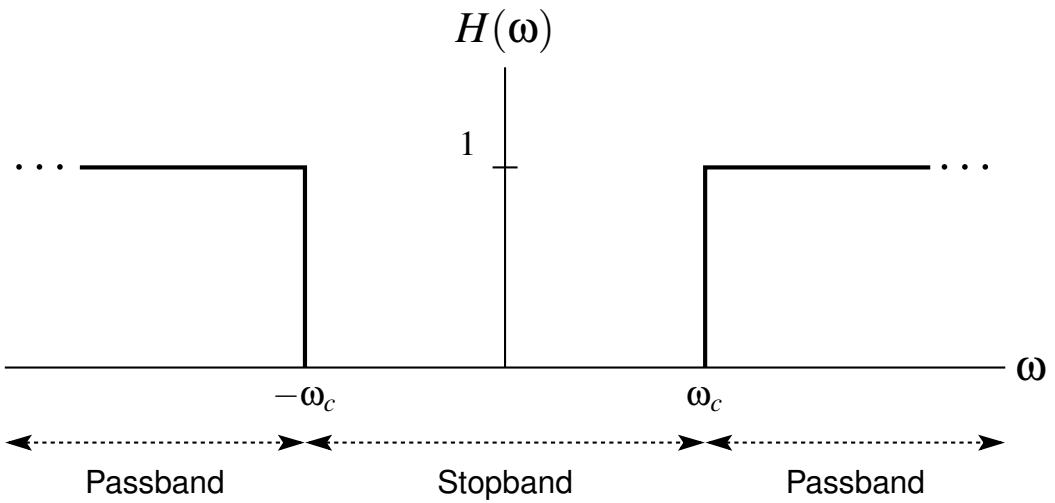


Ideal Highpass Filter

- An **ideal highpass filter** eliminates all frequency components with a frequency whose magnitude is less than some cutoff frequency, while leaving the remaining frequency components unaffected.
- Such a filter has a ***frequency response*** of the form

$$H(\omega) = \begin{cases} 1 & |\omega| \geq \omega_c \\ 0 & \text{otherwise,} \end{cases}$$

  where ω_c is the **cutoff frequency**.
- A plot of this frequency response is given below.

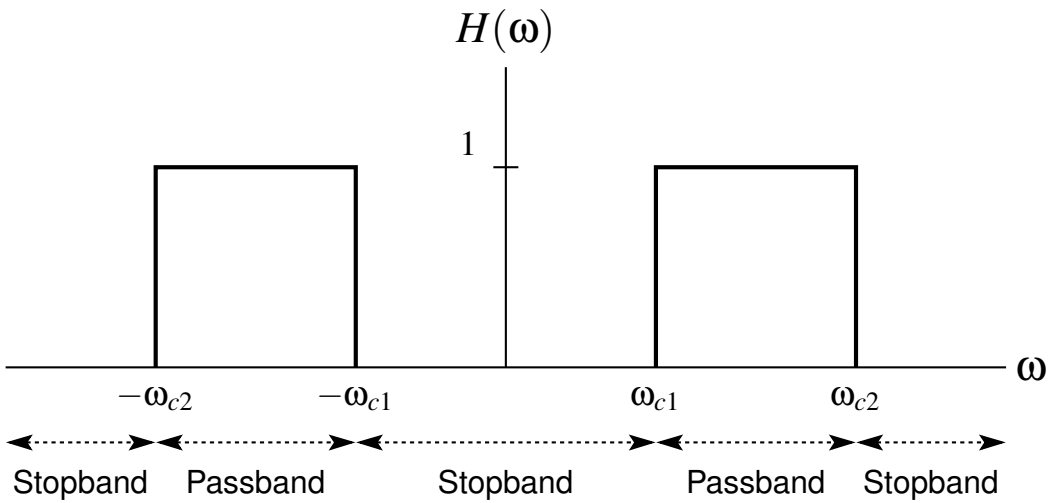


Ideal Bandpass Filter

- An **ideal bandpass filter** eliminates all frequency components with a frequency whose magnitude does not lie in a particular range, while leaving the remaining frequency components unaffected.
- Such a filter has a ***frequency response*** of the form

$$H(\omega) = \begin{cases} 1 & \omega_{c1} \leq |\omega| \leq \omega_{c2} \\ 0 & \text{otherwise,} \end{cases}$$

  where the limits of the passband are ω_{c1} and ω_{c2}.
- A plot of this frequency response is given below.



Part 6

Continuous-Time Fourier Transform (CTFT)

Motivation for the Fourier Transform

- The (CT) Fourier series provide an extremely useful representation for periodic functions.
- Often, however, we need to deal with functions that are not periodic.
- A more general tool than the Fourier series is needed in this case.
- The (CT) Fourier transform can be used to represent both periodic and aperiodic functions.
- Since the Fourier transform is essentially derived from Fourier series through a limiting process, the Fourier transform has many similarities with Fourier series.

Section 6.1

Fourier Transform

Development of the Fourier Transform [Aperiodic Case]

- The (CT) Fourier series is an extremely useful function representation.
- Unfortunately, this function representation can only be used for periodic functions, since a Fourier series is inherently periodic.
- Many functions are not periodic, however.
- Rather than abandoning Fourier series, one might wonder if we can somehow use Fourier series to develop a representation that can be applied to aperiodic functions.
- By viewing an aperiodic function as the limiting case of a T-periodic function where $T \to \infty$, we can use the Fourier series to develop a function representation that can be used for aperiodic functions, known as the Fourier transform.

- Recall that the Fourier series representation of a T-periodic function x is given by

$$x(t) = \sum_{k=-\infty}^{\infty} \underbrace{\left(\frac{1}{T} \int_{-T/2}^{T/2} x(\tau) e^{-jk(2\pi/T)\tau} d\tau \right)}_{c_k} e^{jk(2\pi/T)t}.$$

- In the above representation, if we take the limit as $T \to \infty$, we obtain

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \underbrace{\left(\int_{-\infty}^{\infty} x(\tau) e^{-j\omega\tau} d\tau \right)}_{X(\omega)} e^{j\omega t} d\omega$$

  (i.e., as $T \to \infty$, the outer summation becomes an integral, $\frac{1}{T}$ becomes $\frac{1}{2\pi} d\omega$, and $\left(\frac{2\pi}{T}\right) k$ becomes ω).

- This representation for aperiodic functions is known as the Fourier transform representation.

Generalized Fourier Transform

- The classical Fourier transform for aperiodic functions does not exist (i.e., $\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$ fails to converge) for some functions of great practical interest, such as:
  - a nonzero constant function;
  - a periodic function (e.g., a real or complex sinusoid);
  - the unit-step function (i.e., u); and
  - the signum function (i.e., sgn).
- Fortunately, the Fourier transform can be extended to handle such functions, resulting in what is known as the **generalized Fourier transform**.
- For our purposes, we can think of the classical and generalized Fourier transforms as being defined by the same formulas.
- Therefore, in what follows, we will not typically make a distinction between the classical and generalized Fourier transforms.

CT Fourier Transform (CTFT)

- The (CT) **Fourier transform** of the function x, denoted $\mathcal{F}x$ or X, is given by

$$\mathcal{F}x(\omega) = X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt.$$

- The preceding equation is sometimes referred to as **Fourier transform analysis equation** (or **forward Fourier transform equation**).
- The **inverse Fourier transform** of X, denoted $\mathcal{F}^{-1}X$ or x, is given by

$$\mathcal{F}^{-1}X(t) = x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega.$$

- The preceding equation is sometimes referred to as the **Fourier transform synthesis equation** (or **inverse Fourier transform equation**).
- As a matter of notation, to denote that a function x has the Fourier transform X, we write $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$.
- A function x and its Fourier transform X constitute what is called a **Fourier transform pair**.

Remarks on Operator Notation

- For a function x, the Fourier transform of x is denoted using operator notation as $\mathcal{F}x$.
- The Fourier transform of x evaluated at ω is denoted $\mathcal{F}x(\omega)$.
- Note that $\mathcal{F}x$ is a function, whereas $\mathcal{F}x(\omega)$ is a number.
- Similarly, for a function X, the inverse Fourier transform of X is denoted using operator notation as $\mathcal{F}^{-1}X$.
- The inverse Fourier transform of X evaluated at t is denoted $\mathcal{F}^{-1}X(t)$.
- Note that $\mathcal{F}^{-1}X$ is a function, whereas $\mathcal{F}^{-1}X(t)$ is a number.
- With the above said, engineers often abuse notation, and use expressions like those above to mean things different from their proper meanings.
- Since such notational abuse can lead to problems, it is strongly recommended that one refrain from doing this.

Remarks on Dot Notation

- Often, we would like to write an expression for the Fourier transform of a function without explicitly naming the function.
- For example, consider writing an expression for the Fourier transform of the function $v(t) = x(5t-3)$ but without using the name "v".
- It would be incorrect to write "$\mathcal{F}x(5t-3)$" as this is the function $\mathcal{F}x$ evaluated at $5t-3$, which is not the meaning that we wish to convey.
- Also, strictly speaking, it would be incorrect to write "$\mathcal{F}\{x(5t-3)\}$" as the operand of the Fourier transform operator must be a function, and $x(5t-3)$ is a number (i.e., the function x evaluated at $5t-3$).
- Using dot notation, we can write the following strictly-correct expression for the desired Fourier transform: $\mathcal{F}x(5\cdot-3)$.
- In many cases, however, it is probably advisable to avoid employing anonymous (i.e., unnamed) functions, as their use tends to be more error prone in some contexts.

Remarks on Notational Conventions

- Since dot notation is less frequently used by engineers, the author has elected to minimize its use herein.
- To avoid ambiguous notation, the following conventions are followed:
  1. in the expression for the operand of a Fourier transform operator, the ***independent variable is assumed to be the variable named "t"*** unless otherwise indicated (i.e., in terms of dot notation, each "t" is treated as if it were a "$\cdot$")
  2. in the expression for the operand of the inverse Fourier transform operator, the ***independent variable is assumed to be the variable named "ω"*** unless otherwise indicated (i.e., in terms of dot notation, each "ω" is treated as if it were a "$\cdot$").
- For example, with these conventions:
  - "$\mathcal{F}\{\cos(t-\tau)\}$" denotes the function that is the Fourier transform of the function $v(t)=\cos(t-\tau)$ (not the Fourier transform of the function $v(\tau)=\cos(t-\tau)$).
  - "$\mathcal{F}^{-1}\{\delta(3\omega-\lambda)\}$" denotes the function that is the inverse Fourier transform of the function $V(\omega)=\delta(3\omega-\lambda)$ (not the inverse Fourier transform of the function $V(\lambda)=\delta(3\omega-\lambda)$).

Section 6.2

Convergence Properties of the Fourier Transform

Convergence of the Fourier Transform

- Consider an arbitrary function x.
- The function x has the Fourier transform representation $\tilde{x}$ given by

$$\tilde{x}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega, \quad \text{where} \quad X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt.$$

- Now, we need to concern ourselves with the convergence properties of this representation.
- In other words, we want to know when $\tilde{x}$ is a valid representation of x.
- Since the Fourier transform is essentially derived from Fourier series, the convergence properties of the Fourier transform are closely related to the convergence properties of Fourier series.

Convergence of the Fourier Transform: Continuous Case

- If a function x is ***continuous*** and ***absolutely integrable*** (i.e., $\int_{-\infty}^{\infty} |x(t)|\,dt < \infty$) and the Fourier transform X of x is absolutely integrable (i.e., $\int_{-\infty}^{\infty} |X(\omega)|\,d\omega < \infty$), then the Fourier transform representation of x converges ***pointwise*** (i.e., $x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt\right] e^{j\omega t} d\omega$ for all t).
- Since, in practice, we often encounter functions with discontinuities (e.g., a rectangular pulse), the above result is sometimes of limited value.

Convergence of the Fourier Transform: Finite-Energy Case

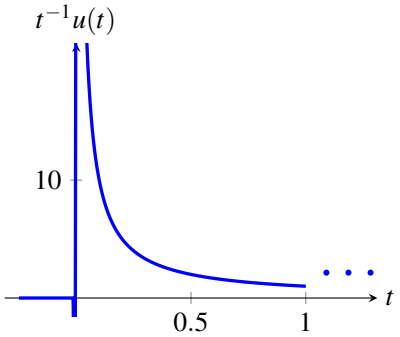
- If a function x is of ***finite energy*** (i.e., $\int_{-\infty}^{\infty} |x(t)|^2 \, dt < \infty$), then its Fourier transform representation converges in the ***MSE sense***.
- In other words, if x is of finite energy, then the energy E in the difference function $\tilde{x} - x$ is zero; that is,

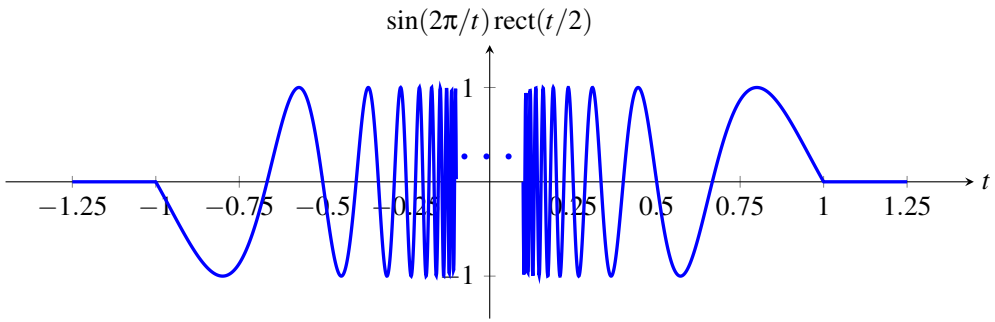
$$E = \int_{-\infty}^{\infty} |\tilde{x}(t) - x(t)|^2 \, dt = 0.$$

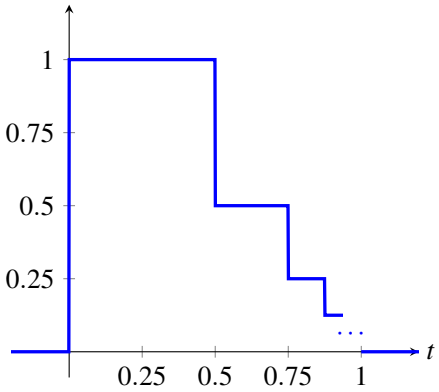
- Since, in situations of practical interest, the finite-energy condition in the above theorem is often satisfied, the theorem is frequently applicable.
- It is important to note, however, that the condition $E = 0$ does not necessarily imply $\tilde{x}(t) = x(t)$ for all t.
- Thus, the above convergence result does not provide much useful information regarding the value of $\tilde{x}(t)$ at specific values of t.
- Consequently, the above theorem is typically most useful for simply determining if the Fourier transform representation converges.

Dirichlet Conditions

- The **Dirichlet conditions** for the function x are as follows:
  1. the function x is ***absolutely integrable*** (i.e., $\int_{-\infty}^{\infty} |x(t)|\,dt < \infty$);
  2. on any finite interval, x has a finite number of maxima and minima (i.e., x is of ***bounded variation***); and
  3. on any finite interval, x has a ***finite number of discontinuities*** and each discontinuity is itself ***finite***.
- Examples of functions violating the Dirichlet conditions are shown below.







Convergence of the Fourier Transform: Dirichlet Case

- If a function x satisfies the ***Dirichlet conditions***, then:
  1. the Fourier transform representation $\tilde{x}$ converges pointwise everywhere to x, except at the points of discontinuity of x; and
  2. at each point t_a of discontinuity of x, the Fourier transform representation $\tilde{x}$ converges to

$$\tilde{x}(t_a) = \frac{1}{2}\left[x(t_a^+) + x(t_a^-)\right],$$

     where $x(t_a^-)$ and $x(t_a^+)$ denote the values of the function x on the left- and right-hand sides of the discontinuity, respectively.
- Since most functions tend to satisfy the Dirichlet conditions and the above convergence result specifies the value of the Fourier transform representation at every point, this result is often very useful in practice.

Section 6.3

Properties of the Fourier Transform

Properties of the (CT) Fourier Transform

Property	Time Domain	Frequency Domain
Linearity	$a_1 x_1(t) + a_2 x_2(t)$	$a_1 X_1(\omega) + a_2 X_2(\omega)$
Time-Domain Shifting	$x(t - t_0)$	$e^{-j\omega t_0} X(\omega)$
Frequency-Domain Shifting	$e^{j\omega_0 t} x(t)$	$X(\omega - \omega_0)$
Time/Frequency-Domain Scaling	$x(at)$	$\frac{1}{\lvert a \rvert} X\left(\frac{\omega}{a}\right)$
Conjugation	$x^*(t)$	$X^*(-\omega)$
Duality	$X(t)$	$2\pi x(-\omega)$
Time-Domain Convolution	$x_1 * x_2(t)$	$X_1(\omega) X_2(\omega)$
Time-Domain Multiplication	$x_1(t) x_2(t)$	$\frac{1}{2\pi} X_1 * X_2(\omega)$
Time-Domain Differentiation	$\frac{d}{dt} x(t)$	$j\omega X(\omega)$
Frequency-Domain Differentiation	$t x(t)$	$j \frac{d}{d\omega} X(\omega)$
Time-Domain Integration	$\int_{-\infty}^{t} x(\tau) d\tau$	$\frac{1}{j\omega} X(\omega) + \pi X(0) \delta(\omega)$

Properties of the (CT) Fourier Transform (Continued)

Property	
Parseval's Relation	$\int_{-\infty}^{\infty} \lvert x(t) \rvert^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} \lvert X(\omega) \rvert^2 d\omega$
Even Symmetry	x is even $\Leftrightarrow$ X is even
Odd Symmetry	x is odd $\Leftrightarrow$ X is odd
Real / Conjugate Symmetry	x is real $\Leftrightarrow$ X is conjugate symmetric

(CT) Fourier Transform Pairs

Pair	$x(t)$	$X(\omega)$
1	$\delta(t)$	1
2	$u(t)$	$\pi\delta(\omega)+\frac{1}{j\omega}$
3	1	$2\pi\delta(\omega)$
4	$\mathrm{sgn}(t)$	$\frac{2}{j\omega}$
5	$e^{j\omega_0 t}$	$2\pi\delta(\omega-\omega_0)$
6	$\cos(\omega_0 t)$	$\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$
7	$\sin(\omega_0 t)$	$\frac{\pi}{j}[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)]$
8	$\mathrm{rect}(t/T)$	$\lvert T\rvert\,\mathrm{sinc}(T\omega/2)$
9	$\frac{\lvert B\rvert}{\pi}\mathrm{sinc}(Bt)$	$\mathrm{rect}\left(\frac{\omega}{2B}\right)$
10	$e^{-at}u(t)$, $\mathrm{Re}\{a\}>0$	$\frac{1}{a+j\omega}$
11	$t^{n-1}e^{-at}u(t)$, $\mathrm{Re}\{a\}>0$	$\frac{(n-1)!}{(a+j\omega)^n}$
12	$\mathrm{tri}(t/T)$	$\frac{\lvert T\rvert}{2}\mathrm{sinc}^2(T\omega/4)$

Linearity

- If $x_1(t) \overset{\text{CTFT}}{\longleftrightarrow} X_1(\omega)$ and $x_2(t) \overset{\text{CTFT}}{\longleftrightarrow} X_2(\omega)$, then

$$a_1 x_1(t) + a_2 x_2(t) \overset{\text{CTFT}}{\longleftrightarrow} a_1 X_1(\omega) + a_2 X_2(\omega),$$

  where a_1 and a_2 are arbitrary complex constants.
- This is known as the **linearity property** of the Fourier transform.

Time-Domain Shifting (Translation)

- If $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$, then

$$x(t - t_0) \overset{\text{CTFT}}{\longleftrightarrow} e^{-j\omega t_0} X(\omega),$$

  where t_0 is an arbitrary real constant.
- This is known as the **translation (or time-domain shifting) property** of the Fourier transform.

Frequency-Domain Shifting (Modulation)

- If $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$, then

$$e^{j\omega_0 t} x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega - \omega_0),$$

  where ω_0 is an arbitrary real constant.
- This is known as the **modulation (or frequency-domain shifting) property** of the Fourier transform.

Time- and Frequency-Domain Scaling (Dilation)

- If $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$, then

$$x(at) \overset{\text{CTFT}}{\longleftrightarrow} \frac{1}{|a|} X\left(\frac{\omega}{a}\right),$$

  where a is an arbitrary nonzero real constant.
- This is known as the **dilation (or time/frequency-domain scaling) property** of the Fourier transform.

Conjugation

- If $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$, then
$$x^*(t) \overset{\text{CTFT}}{\longleftrightarrow} X^*(-\omega).$$
- This is known as the **conjugation property** of the Fourier transform.

Duality

- If $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$, then
$$X(t) \overset{\text{CTFT}}{\longleftrightarrow} 2\pi x(-\omega)$$
- This is known as the **duality property** of the Fourier transform.
- This property follows from the high degree of symmetry in the forward and inverse Fourier transform equations, which are respectively given by
$$X(\lambda) = \int_{-\infty}^{\infty} x(\theta) e^{-j\theta\lambda} d\theta \quad \text{and} \quad x(\lambda) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\theta) e^{j\theta\lambda} d\theta.$$
- That is, the forward and inverse Fourier transform equations are identical except for a ***factor of*** 2π and ***different sign*** in the parameter for the exponential function.
- Although the relationship $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$ only directly provides us with the Fourier transform of $x(t)$, the duality property allows us to indirectly infer the Fourier transform of $X(t)$. Consequently, the duality property can be used to effectively ***double*** the number of Fourier transform pairs that we know.

Time-Domain Convolution

- If $x_1(t) \overset{\text{CTFT}}{\longleftrightarrow} X_1(\omega)$ and $x_2(t) \overset{\text{CTFT}}{\longleftrightarrow} X_2(\omega)$, then

$$x_1 * x_2(t) \overset{\text{CTFT}}{\longleftrightarrow} X_1(\omega)X_2(\omega).$$

- This is known as the **convolution (or time-domain convolution) property** of the Fourier transform.
- In other words, a convolution in the time domain becomes a multiplication in the frequency domain.
- This suggests that the Fourier transform can be used to avoid having to deal with convolution operations.

Time-Domain Multiplication

- If $x_1(t) \overset{\text{CTFT}}{\longleftrightarrow} X_1(\omega)$ and $x_2(t) \overset{\text{CTFT}}{\longleftrightarrow} X_2(\omega)$, then

$$x_1(t)x_2(t) \overset{\text{CTFT}}{\longleftrightarrow} \frac{1}{2\pi} X_1 * X_2(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_1(\theta) X_2(\omega - \theta) d\theta.$$

- This is known as the **(time-domain) multiplication (or frequency-domain convolution) property** of the Fourier transform.
- In other words, multiplication in the time domain becomes convolution in the frequency domain (up to a scale factor of 2π).
- Do not forget the factor of $\frac{1}{2\pi}$ in the above formula!
- This property of the Fourier transform is often tedious to apply (in the forward direction) as it turns a multiplication into a convolution.

Time-Domain Differentiation

- If $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$, then

$$\frac{dx(t)}{dt} \overset{\text{CTFT}}{\longleftrightarrow} j\omega X(\omega).$$

- This is known as the **(time-domain) differentiation property** of the Fourier transform.
- Differentiation in the time domain becomes multiplication by $j\omega$ in the frequency domain.
- Of course, by repeated application of the above property, we have that $\left(\frac{d}{dt}\right)^n x(t) \overset{\text{CTFT}}{\longleftrightarrow} (j\omega)^n X(\omega)$.
- The above suggests that the Fourier transform might be a useful tool when working with differential (or integro-differential) equations.

Frequency-Domain Differentiation

- If $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$, then

$$tx(t) \overset{\text{CTFT}}{\longleftrightarrow} j\frac{d}{d\omega}X(\omega).$$

- This is known as the **frequency-domain differentiation property** of the Fourier transform.

Time-Domain Integration

- If $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$, then

$$\int_{-\infty}^{t} x(\tau)d\tau \overset{\text{CTFT}}{\longleftrightarrow} \frac{1}{j\omega}X(\omega) + \pi X(0)\delta(\omega).$$

- This is known as the **(time-domain) integration property** of the Fourier transform.
- Whereas differentiation in the time domain corresponds to ***multiplication*** by $j\omega$ in the frequency domain, integration in the time domain is associated with ***division*** by $j\omega$ in the frequency domain.
- Since integration in the time domain becomes division by $j\omega$ in the frequency domain, integration can be easier to handle in the frequency domain.
- The above property suggests that the Fourier transform might be a useful tool when working with integral (or integro-differential) equations.

Parseval's Relation

- Recall that the energy of a function x is given by $\int_{-\infty}^{\infty} |x(t)|^2 \, dt$.
- If $x(t) \overset{\text{CTFT}}{\longleftrightarrow} X(\omega)$, then

$$\int_{-\infty}^{\infty} |x(t)|^2 \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 \, d\omega$$

  (i.e., the energy of x and energy of X are equal up to a factor of 2π).
- This relationship is known as **Parseval's relation**.
- Since energy is often a quantity of great significance in engineering applications, it is extremely helpful to know that the Fourier transform ***preserves energy*** (up to a scale factor).

Even/Odd Symmetry

- For a function x with Fourier transform X, the following assertions hold:

$$x \text{ is even} \Leftrightarrow X \text{ is even}; \quad \text{and}$$
$$x \text{ is odd} \Leftrightarrow X \text{ is odd}.$$

- In other words, the forward and inverse Fourier transforms preserve even/odd symmetry.

- A function x is ***real*** if and only if its Fourier transform X satisfies

$$X(\omega) = X^*(-\omega) \text{ for all } \omega$$

(i.e., X is ***conjugate symmetric***).

- Thus, for a real-valued function, the portion of the graph of $X(\omega)$ for $\omega < 0$ is ***completely redundant***, as it is determined by symmetry.
- From properties of complex numbers, one can show that $X(\omega) = X^*(-\omega)$ is equivalent to

$$|X(\omega)| = |X(-\omega)| \quad \text{and} \quad \arg X(\omega) = -\arg X(-\omega)$$

(i.e., $|X(\omega)|$ is ***even*** and $\arg X(\omega)$ is ***odd***).

- Note that x being real does ***not*** necessarily imply that X is real.

THIS SLIDE IS INTENTIONALLY LEFT BLANK.

Section 6.4

Fourier Transform of Periodic Functions

Fourier Transform of Periodic Functions

- The Fourier transform can be generalized to also handle periodic functions.
- Consider a periodic function x with period T and frequency $\omega_0 = \frac{2\pi}{T}$.
- Define the function x_T as

$$x_T(t) = \begin{cases} x(t) & -\frac{T}{2} \leq t < \frac{T}{2} \\ 0 & \text{otherwise.} \end{cases}$$

  (i.e., $x_T(t)$ is equal to $x(t)$ over a single period and zero elsewhere).
- Let a denote the Fourier series coefficient sequence of x.
- Let X and X_T denote the Fourier transforms of x and x_T, respectively.
- The following relationships can be shown to hold:

$$X(\omega) = \sum_{k=-\infty}^{\infty} \omega_0 X_T(k\omega_0)\delta(\omega - k\omega_0),$$

$$a_k = \tfrac{1}{T} X_T(k\omega_0), \quad \text{and} \quad X(\omega) = \sum_{k=-\infty}^{\infty} 2\pi a_k \delta(\omega - k\omega_0).$$

Fourier Transform of Periodic Functions (Continued)

- The Fourier transform X of a periodic function is a series of impulses that occur at integer multiples of the fundamental frequency ω_0 (i.e., $X(\omega) = \sum_{k=-\infty}^{\infty} 2\pi a_k \delta(\omega - k\omega_0)$).
- Due to the preceding fact, the Fourier transform of a periodic function can only be nonzero at integer multiples of the fundamental frequency.
- The Fourier series coefficient sequence a is produced by sampling X_T at integer multiples of the fundamental frequency ω_0 and scaling the resulting sequence by $\frac{1}{T}$ (i.e., $a_k = \frac{1}{T} X_T(k\omega_0)$).

Section 6.5

Fourier Transform and Frequency Spectra of Functions

The Frequency-Domain Perspective on Functions

- Like Fourier series, the Fourier transform also provides us with a frequency-domain perspective on functions.
- That is, instead of viewing a function as having information distributed with respect to ***time*** (i.e., a function whose domain is time), we view a function as having information distributed with respect to ***frequency*** (i.e., a function whose domain is frequency).
- The Fourier transform of a function x provides a means to ***quantify*** how much information x has at different frequencies.
- The distribution of information in a function over different frequencies is referred to as the ***frequency spectrum*** of the function.

Fourier Transform and Frequency Spectra

- To gain further insight into the role played by the Fourier transform X in the context of the frequency spectrum of x, it is helpful to write the Fourier transform representation of x with $X(\omega)$ expressed in ***polar form*** as follows:

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(\omega)|\, e^{j[\omega t + \arg X(\omega)]}d\omega.$$

- In effect, the quantity $|X(\omega)|$ is a ***weight*** that determines how much the complex sinusoid at frequency ω contributes to the integration result x.
- The quantity $\arg X(\omega)$ determines how the complex sinusoid at frequency ω is shifted related to complex sinusoids at other frequencies.
- Perhaps, this can be more easily seen if we express the above integral as the ***limit of a sum***, derived from an approximation of the integral using the areas of rectangles, as shown on the next slide. [Recall that $\int_{-\infty}^{\infty} f(x)dx = \lim_{\Delta x \to 0} \sum_{k=-\infty}^{\infty} \Delta x f(k\Delta x)$.]

Fourier Transform and Frequency Spectra (Continued 1)

- Expressing the integral (from the previous slide) as the ***limit of a sum***, we obtain

$$x(t) = \lim_{\Delta\omega \to 0} \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \Delta\omega \, |X(\omega)| \, e^{j[\omega t + \arg X(\omega)]},$$

  where $\omega = k\Delta\omega$.

- In the above equation, the kth term in the summation corresponds to a complex sinusoid with fundamental frequency $\omega = k\Delta\omega$ that has had its ***amplitude scaled*** by a factor of $|X(\omega)|$ and has been ***time shifted*** by an amount that depends on $\arg X(\omega)$.
- For a given $\omega = k\Delta\omega$ (which is associated with the kth term in the summation), the ***larger*** $|X(\omega)|$ is, the larger the amplitude of its corresponding complex sinusoid $e^{j\omega t}$ will be, and therefore the ***larger the contribution*** the kth term will make to the overall summation.
- In this way, we can use $|X(\omega)|$ as a ***measure*** of how much information a function x has at the frequency ω.

Fourier Transform and Frequency Spectra (Continued 2)

- The Fourier transform X of the function x is referred to as the **frequency spectrum** of x.
- The magnitude $|X(\omega)|$ of the Fourier transform X is referred to as the **magnitude spectrum** of x.
- The argument $\arg X(\omega)$ of the Fourier transform X is referred to as the **phase spectrum** of x.
- Since the Fourier transform is a function of a real variable, a function can potentially have information at any real frequency.
- Since the Fourier transform X of a periodic function x with fundamental frequency ω_0 and the Fourier series coefficient sequence a is given by $X(\omega) = \sum_{k=-\infty}^{\infty} 2\pi a_k \delta(\omega - k\omega_0)$, the Fourier transform and Fourier series give consistent results for the frequency spectrum of a periodic function.
- Since the frequency spectrum is complex (in the general case), it is ***usually represented using two plots***, one showing the magnitude spectrum and one showing the phase spectrum.

Frequency Spectra of Real Functions

- Recall that, for a real function x, the Fourier transform X of x satisfies

$$X(\omega) = X^*(-\omega)$$

(i.e., X is ***conjugate symmetric***), which is equivalent to

$$|X(\omega)| = |X(-\omega)| \quad \text{and} \quad \arg X(\omega) = -\arg X(-\omega).$$

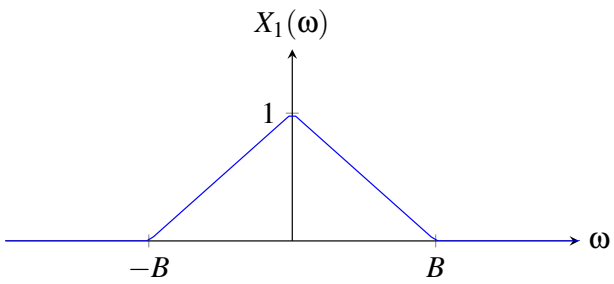
- Since $|X(\omega)| = |X(-\omega)|$, the magnitude spectrum of a real function is always ***even***.
- Similarly, since $\arg X(\omega) = -\arg X(-\omega)$, the phase spectrum of a real function is always ***odd***.
- Due to the symmetry in the frequency spectra of real functions, we typically ***ignore negative frequencies*** when dealing with such functions.
- In the case of functions that are complex but not real, frequency spectra do not possess the above symmetry, and ***negative frequencies become important***.

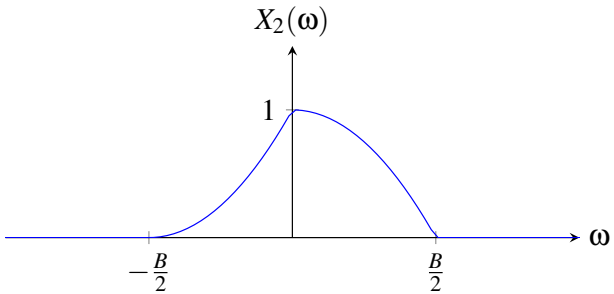
Bandwidth

- A function with the Fourier transform X is said to be **bandlimited** if, for some (finite) nonnegative real constant B, the following condition holds:

$$X(\omega) = 0 \text{ for all } \omega \text{ satisfying } |\omega| > B.$$

- The **bandwidth** B of a function with the Fourier transform X is defined as $B = \omega_1 - \omega_0$, where $X(\omega) = 0$ for all $\omega \notin [\omega_0, \omega_1]$.
- In the case of ***real-valued*** functions, however, this definition of bandwidth is usually amended to consider ***only nonnegative*** frequencies.
- The real-valued function x_1 and complex-valued function x_2 with the respective Fourier transforms X_1 and X_2 shown below each have bandwidth B (where only nonnegative frequencies are considered in the case of x_1).





- One can show that a function ***cannot be both time limited and bandlimited***.

Energy-Density Spectra

- By Parseval's relation, the energy E in a function x with Fourier transform X is given by

$$E = \frac{1}{2\pi} \int_{-\infty}^{\infty} E_x(\omega) d\omega,$$

where

$$E_x(\omega) = |X(\omega)|^2.$$

- We refer to E_x as the **energy-density spectrum** of the function x.
- The function E_x indicates how the energy in x is distributed with respect to frequency.
- For example, the energy contributed by frequencies in the range $[\omega_1, \omega_2]$ is given by

$$\frac{1}{2\pi} \int_{\omega_1}^{\omega_2} E_x(\omega) d\omega.$$

Section 6.6

Fourier Transform and LTI Systems

Frequency Response of LTI Systems

- Consider a LTI system with input x, output y, and impulse response h, and let X, Y, and H denote the Fourier transforms of x, y, and h, respectively.
- Since $y(t) = x * h(t)$, we have that

$$Y(\omega) = X(\omega)H(\omega).$$

- The function H is called the **frequency response** of the system.
- A LTI system is ***completely characterized*** by its frequency response H.
- The above equation provides an alternative way of viewing the behavior of a LTI system. That is, we can view the system as operating in the frequency domain on the Fourier transforms of the input and output functions.
- The frequency spectrum of the output is the product of the frequency spectrum of the input and the frequency response of the system.

Frequency Response of LTI Systems (Continued 1)

- In the general case, the frequency response H is a complex-valued function.
- Often, we represent $H(\omega)$ in terms of its magnitude $|H(\omega)|$ and argument $\arg H(\omega)$.
- The quantity $|H(\omega)|$ is called the **magnitude response** of the system.
- The quantity $\arg H(\omega)$ is called the **phase response** of the system.
- Since $Y(\omega) = X(\omega)H(\omega)$, we trivially have that

$$|Y(\omega)| = |X(\omega)|\,|H(\omega)| \quad \text{and} \quad \arg Y(\omega) = \arg X(\omega) + \arg H(\omega).$$

- The magnitude spectrum of the output equals the magnitude spectrum of the input times the magnitude response of the system.
- The phase spectrum of the output equals the phase spectrum of the input plus the phase response of the system.

Frequency Response of LTI Systems (Continued 2)

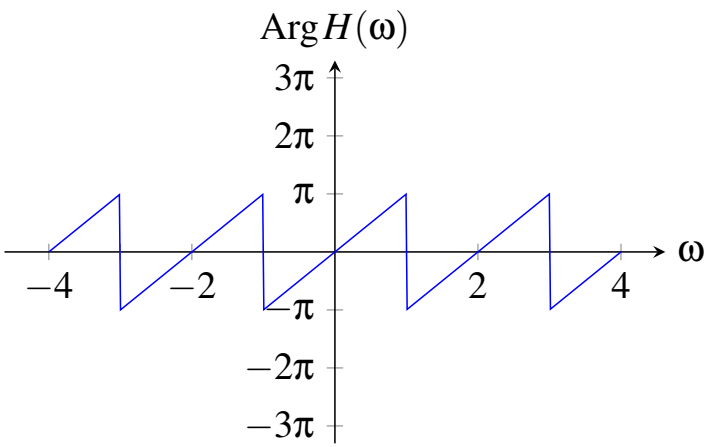
- Since the frequency response H is simply the frequency spectrum of the impulse response h, if h is ***real***, then

$$|H(\omega)| = |H(-\omega)| \quad \text{and} \quad \arg H(\omega) = -\arg H(-\omega)$$

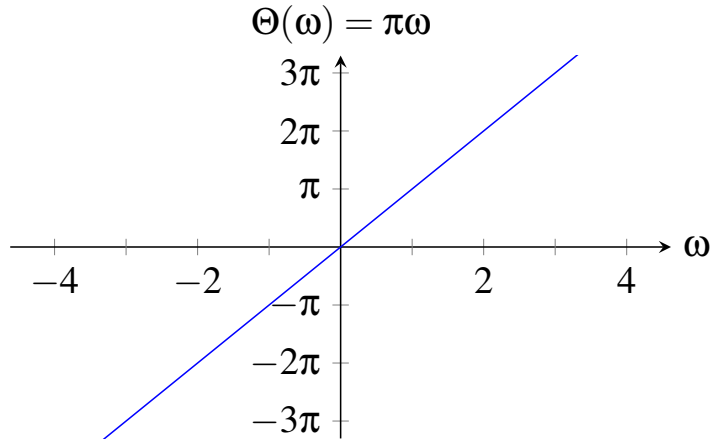
(i.e., the magnitude response $|H(\omega)|$ is ***even*** and the phase response $\arg H(\omega)$ is ***odd***).

Unwrapped Phase

- For many types of analysis, restricting the range of a phase function to an interval of length 2π (such as $(-\pi, \pi]$), often unnecessarily introduces discontinuities into the function.
- This motivates the notion of unwrapped phase.
- The **unwrapped phase** is simply the phase defined in such a way so as not to restrict the phase to an interval of length 2π and to keep the phase function continuous to the greatest extent possible.
- For example, the function $H(\omega) = e^{j\pi\omega}$ has the unwrapped phase $\Theta(\omega) = \pi\omega$.



Phase



Unwrapped Phase

Interpretation of Magnitude and Phase Response

- Recall that a LTI system $\mathcal{H}$ with frequency response H is such that

$$\mathcal{H}\left\{e^{j\omega t}\right\}(t) = H(\omega)e^{j\omega t}.$$

- Expressing $H(\omega)$ in polar form, we have

$$\begin{aligned}\mathcal{H}\{e^{j\omega t}\}(t) &= |H(\omega)|\, e^{j\arg H(\omega)} e^{j\omega t} \\ &= |H(\omega)|\, e^{j[\omega t + \arg H(\omega)]} \\ &= |H(\omega)|\, e^{j\omega(t+\arg[H(\omega)]/\omega)}.\end{aligned}$$

- Thus, the response of the system to the function $e^{j\omega t}$ is produced by applying two transformations to this function:
  - (amplitude) scaling by $|H(\omega)|$; and
  - translating by $-\frac{\arg H(\omega)}{\omega}$.
- Therefore, the magnitude response determines how different complex sinusoids are ***scaled*** (in amplitude) by the system.
- Similarly, the phase response determines how different complex sinusoids are ***translated*** (i.e., delayed/advanced) by the system.

Magnitude Distortion

- Recall that a LTI system $\mathcal{H}$ with frequency response H is such that

$$\mathcal{H}\{e^{j\omega t}\}(t) = |H(\omega)|\, e^{j\omega(t+\arg[H(\omega)]/\omega)}.$$

- If $|H(\omega)|$ is a constant (for all ω), every complex sinusoid is scaled by the same amount when passing through the system.
- A system for which $|H(\omega)| = 1$ (for all ω) is said to be **allpass**.
- In the case of an allpass system, the magnitude spectra of the system's input and output are identical.
- If $|H(\omega)|$ is not a constant, different complex sinusoids are scaled by different amounts, resulting in what is known as **magnitude distortion**.

Phase Distortion

- Recall that a LTI system $\mathcal{H}$ with frequency response H is such that

$$\mathcal{H}\{e^{j\omega t}\}(t) = |H(\omega)|\, e^{j\omega(t+\arg[H(\omega)]/\omega)}.$$

- The preceding equation can be rewritten as

$$\mathcal{H}\{e^{j\omega t}\}(t) = |H(\omega)|\, e^{j\omega[t-\tau_p(\omega)]} \quad \text{where} \quad \tau_p(\omega) = -\frac{\arg H(\omega)}{\omega}.$$

- The function τ_p is known as the **phase delay** of the system.
- If $\tau_p(\omega) = t_d$ (where t_d is a constant), the system shifts all complex sinusoids by the same amount t_d.
- Since $\tau_p(\omega) = t_d$ is equivalent to the (unwrapped) phase response being of the form $\arg H(\omega) = -t_d\omega$ (which is a linear function with a zero constant term), a system with a constant phase delay is said to have **linear phase**.
- In the case that $\tau_p(\omega) = 0$, the system is said to have **zero phase**.
- If $\tau_p(\omega)$ is not a constant, different complex sinusoids are shifted by different amounts, resulting in what is known as **phase distortion**.

Distortionless Transmission

- Consider a LTI system $\mathcal{H}$ with input x and output y given by

$$y(t) = x(t - t_0),$$

  where t_0 is a real constant.
- That is, the output of the system is simply the input delayed by t_0.
- This type of behavior is the ideal for which we strive in real-world communication systems (i.e., the received signal y equals a delayed version of the transmitted signal x).
- Taking the Fourier transform of the preceding equation, we have

$$Y(\omega) = e^{-j\omega t_0} X(\omega).$$

- Thus, the system has the frequency response H given by

$$H(\omega) = e^{-j\omega t_0}.$$

- Since the phase delay of the system is $\tau_p(\omega) = -\left(\frac{-\omega t_0}{\omega}\right) = t_0$, the phase delay is constant and the system has linear phase.

Magnitude and Phase Distortion in Audio

- The relative importance of the magnitude spectrum and phase spectrum is highly dependent on the particular application of interest.
- Consider the case of the human auditory system (i.e., human hearing).
- The human auditory system tends to be quite sensitive to changes in the magnitude spectrum of an audio signal.
- That is, a significant change in the magnitude spectrum of an audio signal is very likely to lead to a noticeable difference in the perceived sound.
- On the other hand, the human auditory system tends to be much less sensitive to changes in the phase spectrum of an audio signal.
- In other words, changes to the phase spectrum of an audio signal are often only barely perceptible or not perceptible at all.
- For the above reasons, in applications involving the human auditory system, magnitude distortion often tends to be more of a concern than phase distortion.

Magnitude and Phase Distortion in Images

- Consider the case of the human visual system.
- The human visual system tends to be quite sensitive to changes in the phase spectrum of an image.
- That is, a significant change in the phase spectrum of an image is likely to lead to a very substantial difference in how the image is perceived.
- The phase spectrum of an image tends to capture information about the location of the edges in the image, and edges are play a crucial role in how humans perceive images.
- On the other hand, the human visual system tends to be somewhat less sensitive to changes in the magnitude spectrum of an image.
- For the above reasons, phase distortion is usually deemed highly undesirable in systems that process images, when the image data is to be consumed by humans.

Example: Magnitude and Phase Distortion in Images (1)

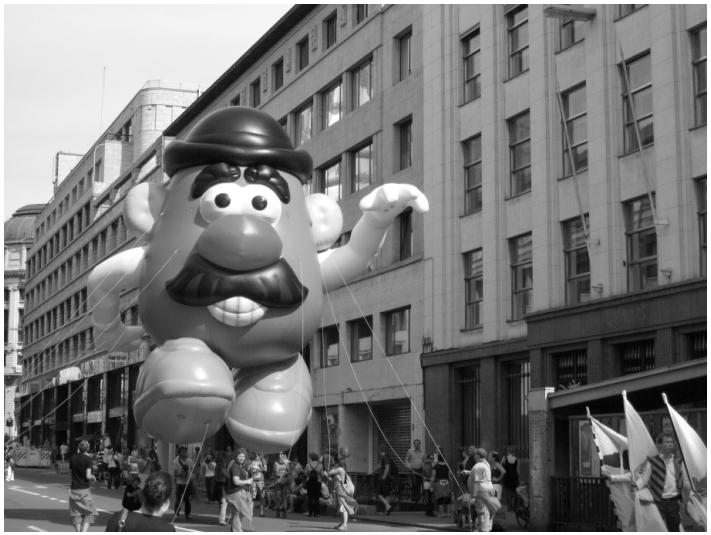
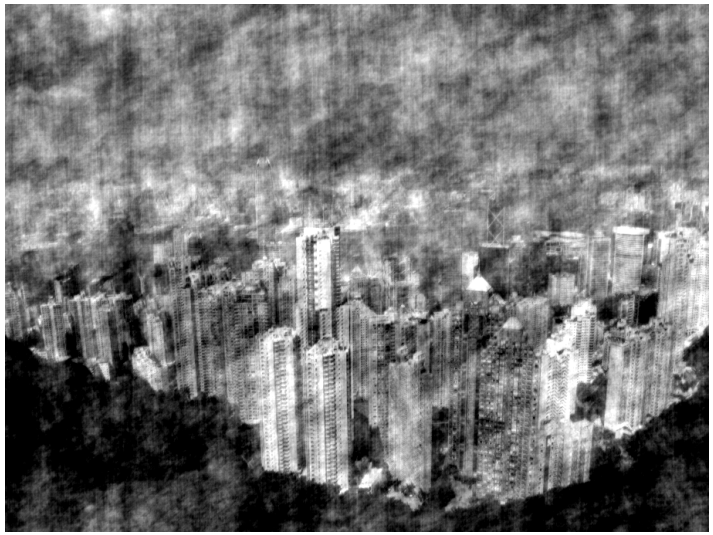


Image A

Image B

Magnitude Spectrum from Image B and
Phase Spectrum from Image A



Magnitude Spectrum from Image A and
Phase Spectrum from Image B

Example: Magnitude and Phase Distortion in Images (2)

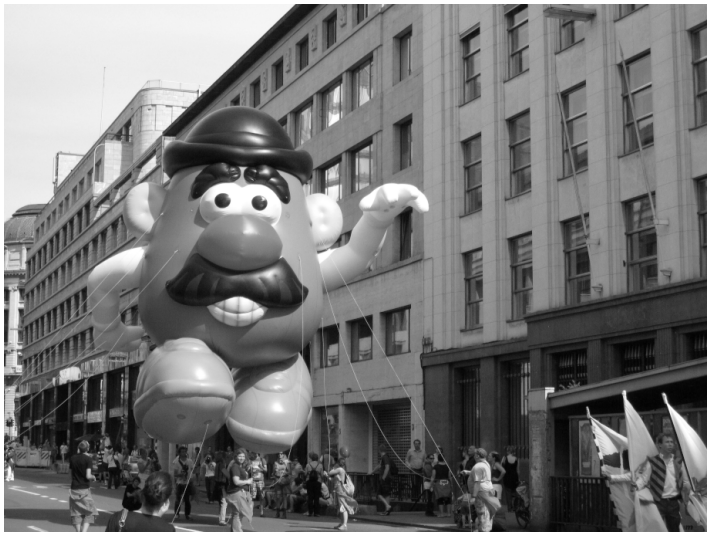


Image A

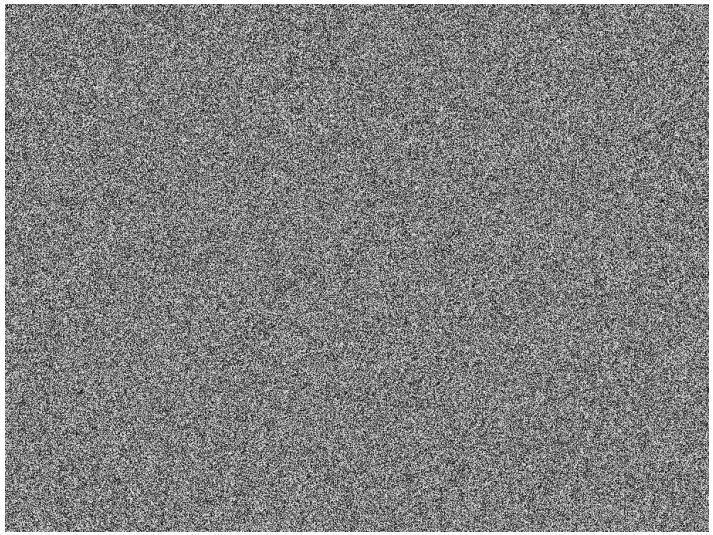
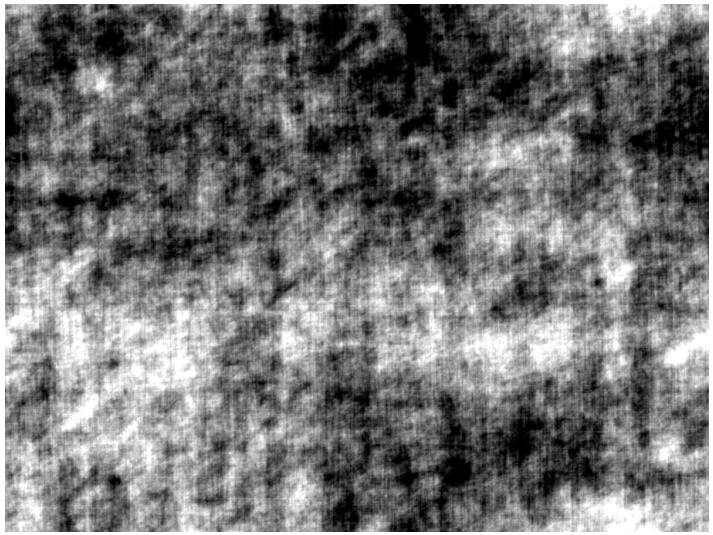


Image B (White Noise)

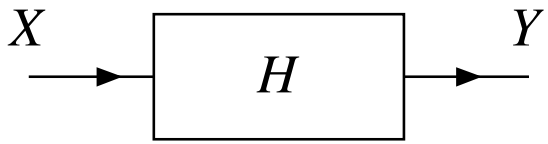
Magnitude Spectrum from Image B and
Phase Spectrum from Image A



Magnitude Spectrum from Image A and
Phase Spectrum from Image B

Block Diagram Representations of LTI Systems

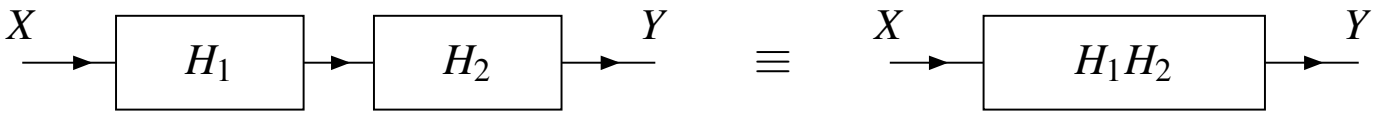
- Consider a LTI system with input x, output y, and impulse response h, and let X, Y, and H denote the Fourier transforms of x, y, and h, respectively.
- Often, it is convenient to represent such a system in block diagram form in the frequency domain as shown below.



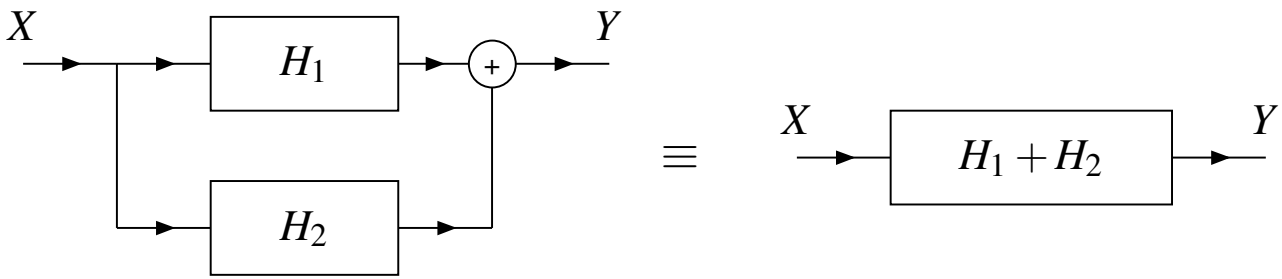
- Since a LTI system is completely characterized by its frequency response, we typically label the system with this quantity.

Interconnection of LTI Systems

- The ***series*** interconnection of the LTI systems with frequency responses H_1 and H_2 is the LTI system with frequency response $H_1 H_2$. That is, we have the equivalence shown below.



- The ***parallel*** interconnection of the LTI systems with frequency responses H_1 and H_2 is the LTI system with the frequency response $H_1 + H_2$. That is, we have the equivalence shown below.



LTI Systems and Differential Equations

- Many LTI systems of practical interest can be represented using an ***Nth-order linear differential equation with constant coefficients***.
- Consider a system with input x and output y that is characterized by an equation of the form

$$\sum_{k=0}^{N} b_k \left(\frac{d}{dt}\right)^k y(t) = \sum_{k=0}^{M} a_k \left(\frac{d}{dt}\right)^k x(t),$$

  where the a_k and b_k are complex constants and $M \leq N$.
- Let h denote the impulse response of the system, and let X, Y, and H denote the Fourier transforms of x, y, and h, respectively.
- One can show that H is given by

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{\sum_{k=0}^{M} a_k j^k \omega^k}{\sum_{k=0}^{N} b_k j^k \omega^k}.$$

- Observe that, for a system of the form considered above, the frequency response is a ***rational function***.

Section 6.7

Application: Filtering

Filtering

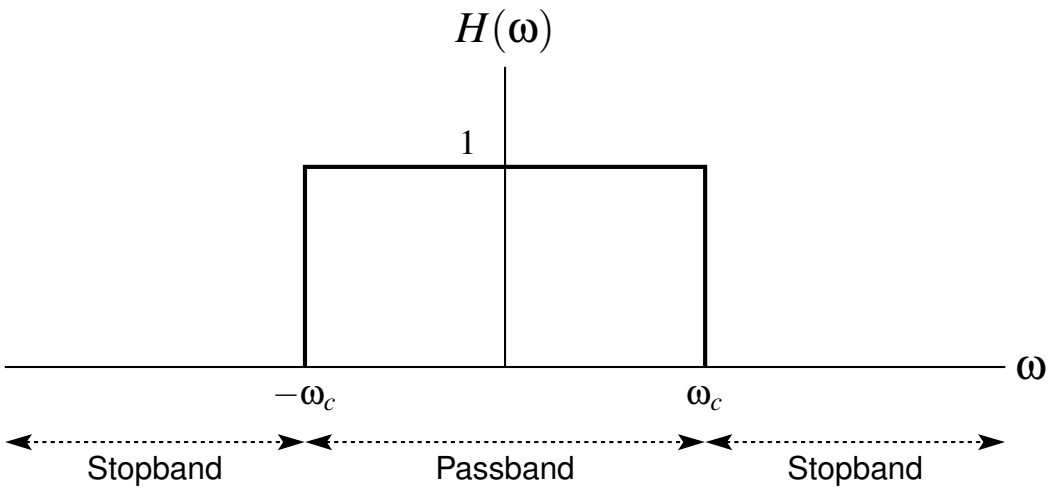
- In many applications, we want to ***modify the spectrum*** of a function by either amplifying or attenuating certain frequency components.
- This process of modifying the frequency spectrum of a function is called **filtering**.
- A system that performs a filtering operation is called a **filter**.
- Many types of filters exist.
- **Frequency selective filters** pass some frequencies with little or no distortion, while significantly attenuating other frequencies.
- Several basic types of frequency-selective filters include: lowpass, highpass, and bandpass.

Ideal Lowpass Filter

- An **ideal lowpass filter** eliminates all frequency components with a frequency whose magnitude is greater than some cutoff frequency, while leaving the remaining frequency components unaffected.
- Such a filter has a ***frequency response*** H of the form

$$H(\omega) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & \text{otherwise,} \end{cases}$$

  where ω_c is the **cutoff frequency**.
- A plot of this frequency response is given below.

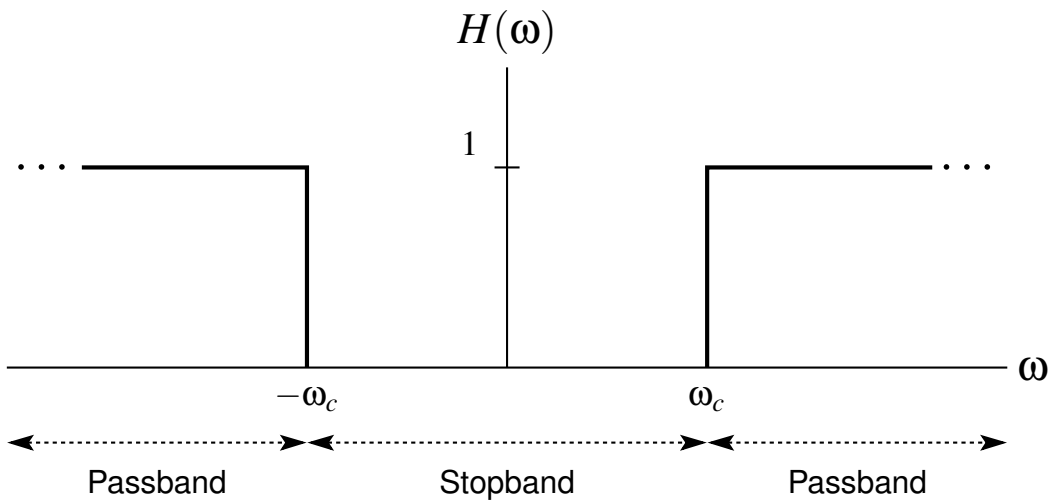


Ideal Highpass Filter

- An **ideal highpass filter** eliminates all frequency components with a frequency whose magnitude is less than some cutoff frequency, while leaving the remaining frequency components unaffected.
- Such a filter has a ***frequency response*** H of the form

$$H(\omega) = \begin{cases} 1 & |\omega| \geq \omega_c \\ 0 & \text{otherwise,} \end{cases}$$

  where ω_c is the **cutoff frequency**.
- A plot of this frequency response is given below.

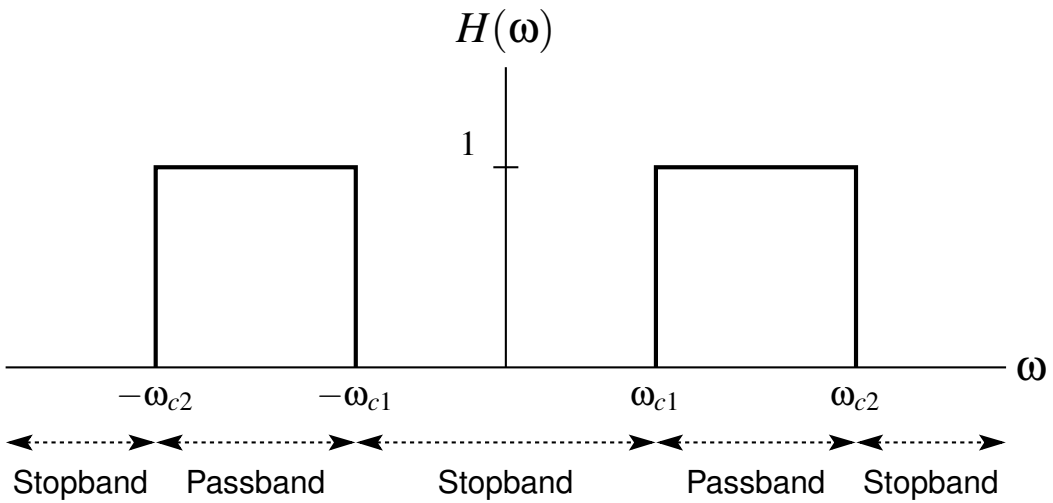


Ideal Bandpass Filter

- An **ideal bandpass filter** eliminates all frequency components with a frequency whose magnitude does not lie in a particular range, while leaving the remaining frequency components unaffected.
- Such a filter has a ***frequency response*** H of the form

$$H(\omega) = \begin{cases} 1 & \omega_{c1} \leq |\omega| \leq \omega_{c2} \\ 0 & \text{otherwise,} \end{cases}$$

  where the limits of the passband are ω_{c1} and ω_{c2}.
- A plot of this frequency response is given below.



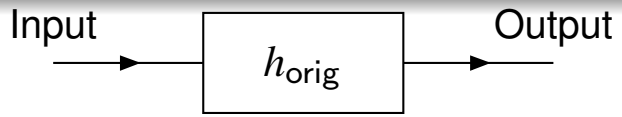
Section 6.8

Application: Equalization

Equalization

- Often, we find ourselves faced with a situation where we have a system with a particular frequency response that is undesirable for the application at hand.
- As a result, we would like to change the frequency response of the system to be something more desirable.
- This process of modifying the frequency response in this way is referred to as **equalization**. [Essentially, equalization is just a filtering operation.]
- Equalization is used in many applications.
- In real-world ***communication systems***, equalization is used to eliminate or minimize the distortion introduced when a signal is sent over a (nonideal) communication channel.
- In ***audio applications***, equalization can be employed to emphasize or de-emphasize certain ranges of frequencies. For example, equalization can be used to boost the bass (i.e., emphasize the low frequencies) in the audio output of a stereo.

Equalization (Continued)



Original System

New System with Equalization

- Let H_{orig} denote the frequency response of ***original*** system (i.e., without equalization).
- Let H_{d} denote the ***desired*** frequency response.
- Let H_{eq} denote the frequency response of the ***equalizer***.
- The new system with equalization has frequency response

$$H_{\text{new}}(\omega) = H_{\text{eq}}(\omega)H_{\text{orig}}(\omega).$$

- By choosing $H_{\text{eq}}(\omega) = H_{\text{d}}(\omega)/H_{\text{orig}}(\omega)$, the new system with equalization will have the frequency response

$$H_{\text{new}}(\omega) = [H_{\text{d}}(\omega)/H_{\text{orig}}(\omega)]\,H_{\text{orig}}(\omega) = H_{\text{d}}(\omega).$$

- In effect, by using an equalizer, we can obtain a new system with the frequency response that we desire.

Section 6.9

Application: Circuit Analysis

Electronic Circuits

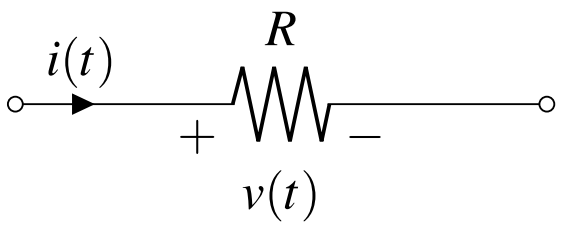
- An **electronic circuit** is a network of one or more interconnected circuit elements.
- The three most basic types of circuit elements are:
  1. resistors;
  2. inductors; and
  3. capacitors.
- Two fundamental quantities of interest in electronic circuits are current and voltage.
- **Current** is the rate at which electric charge flows through some part of a circuit, such as a circuit element, and is measured in units of amperes (A).
- **Voltage** is the difference in electric potential between two points in a circuit, such as across a circuit element, and is measured in units of volts (V).
- Voltage is essentially a force that makes electric charge (or current) flow.

Resistors

- A **resistor** is a circuit element that opposes the flow of current.
- A resistor is characterized by an equation of the form

$$v(t) = Ri(t) \quad \left(\text{or equivalently, } i(t) = \tfrac{1}{R}v(t)\right),$$

  where R is a nonnegative real constant, and v and i respectively denote the voltage across and current through the resistor as a function of time.
- As a matter of terminology, the quantity R is known as the **resistance** of the resistor.
- Resistance is measured in units of ohms (Ω).
- In circuit diagrams, a resistor is denoted by the symbol shown below.

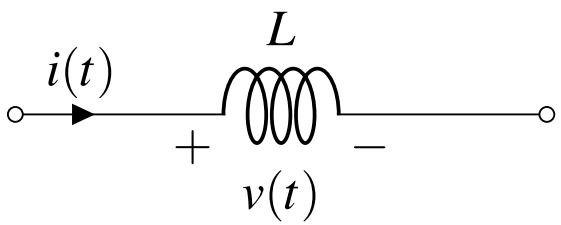


Inductors

- An **inductor** is a circuit element that converts an electric current into a magnetic field and vice versa.
- An inductor uses the energy stored in a magnetic field in order to ***oppose changes in current*** (through the inductor).
- An inductor is characterized by an equation of the form

$$v(t) = L\frac{d}{dt}i(t) \quad \text{(or equivalently, } i(t) = \frac{1}{L}\int_{-\infty}^{t} v(\tau)d\tau\text{)},$$

  where L is a nonnegative real constant, and v and i respectively denote the voltage across and current through the inductor as a function of time.
- As a matter of terminology, the quantity L is known as the **inductance** of the inductor.
- Inductance is measured in units of henrys (H).
- In circuit diagrams, an inductor is denoted by the symbol shown below.

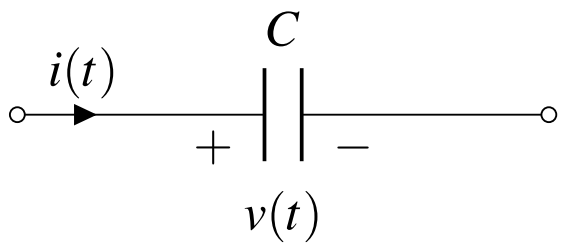


Capacitors

- A **capacitor** is a circuit element that stores electric charge.
- A capacitor uses the energy stored in an electric field in order to ***oppose changes in voltage*** (across the capacitor).
- A capacitor is characterized by an equation of the form

$$v(t) = \frac{1}{C}\int_{-\infty}^{t} i(\tau)d\tau \quad \text{(or equivalently, } i(t) = C\frac{d}{dt}v(t)\text{)},$$

  where C is a nonnegative real constant, and v and i respectively denote the voltage across and current through the capacitor as a function of time.
- As a matter of terminology, the quantity C is known as the **capacitance** of the capacitor.
- Capacitance is measured in units of farads (F).
- In circuit diagrams, a capacitor is denoted by the symbol shown below.



Circuit Analysis with the Fourier Transform

- The Fourier transform is a very useful tool for circuit analysis.
- The utility of the Fourier transform is partly due to the fact that the ***differential/integral*** equations that describe inductors and capacitors are much simpler to express in the Fourier domain than in the time domain.
- Let v and i denote the voltage across and current through a circuit element, and let V and I denote the Fourier transforms of v and i, respectively.
- In the frequency domain, the equations characterizing a resistor, an inductor, and a capacitor respectively become:

$$V(\omega) = RI(\omega) \quad \text{(or equivalently, } I(\omega) = \tfrac{1}{R}V(\omega)\text{);}$$
$$V(\omega) = j\omega LI(\omega) \quad \text{(or equivalently, } I(\omega) = \tfrac{1}{j\omega L}V(\omega)\text{);} \quad \text{and}$$
$$V(\omega) = \tfrac{1}{j\omega C}I(\omega) \quad \text{(or equivalently, } I(\omega) = j\omega CV(\omega)\text{).}$$

- Note the absence of differentiation and integration in the above equations for an inductor and a capacitor.

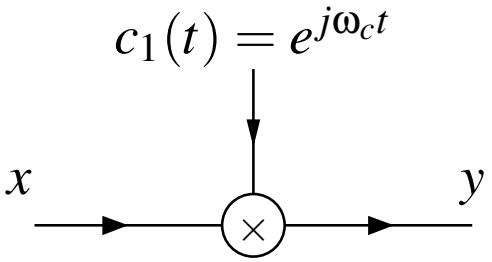
Section 6.10

Application: Amplitude Modulation (AM)

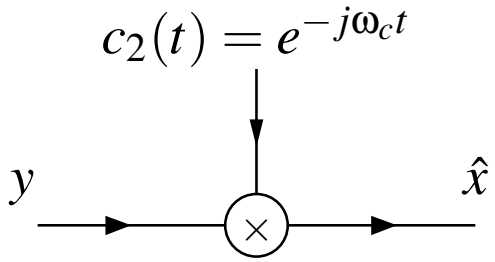
Motivation for Amplitude Modulation (AM)

- In communication systems, we often need to transmit a signal using a frequency range that is different from that of the original signal.
- For example, voice/audio signals typically have information in the range of 0 to 22 kHz.
- Often, it is not practical to transmit such a signal using its original frequency range.
- Two potential problems with such an approach are:
  1. interference; and
  2. constraints on antenna length.
- Since many signals are broadcast over the airwaves, we need to ensure that no two transmitters use the same frequency bands in order to avoid interference.
- Also, in the case of transmission via electromagnetic waves (e.g., radio waves), the length of antenna required becomes impractically large for the transmission of relatively low frequency signals.
- For the preceding reasons, we often need to change the frequency range associated with a signal before transmission.

Trivial Amplitude Modulation (AM) System



Transmitter



Receiver

- The transmitter is characterized by

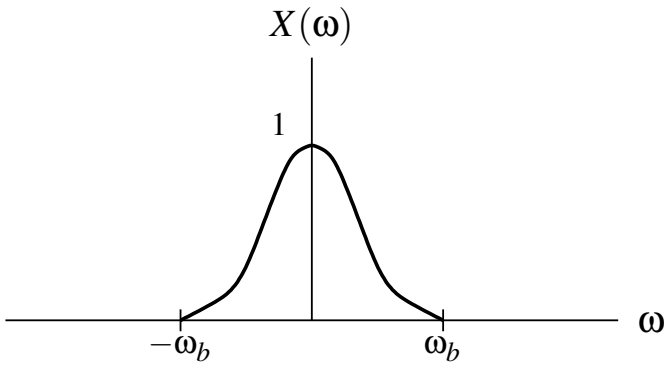
$$y(t) = e^{j\omega_c t} x(t) \quad \Longleftrightarrow \quad Y(\omega) = X(\omega - \omega_c).$$

- The receiver is characterized by

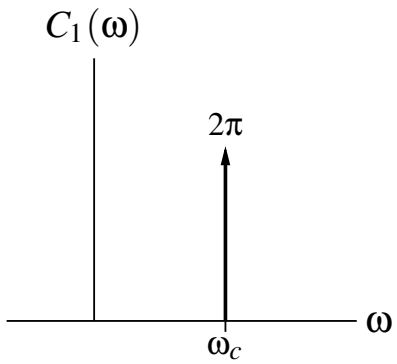
$$\hat{x}(t) = e^{-j\omega_c t} y(t) \quad \Longleftrightarrow \quad \hat{X}(\omega) = Y(\omega + \omega_c).$$

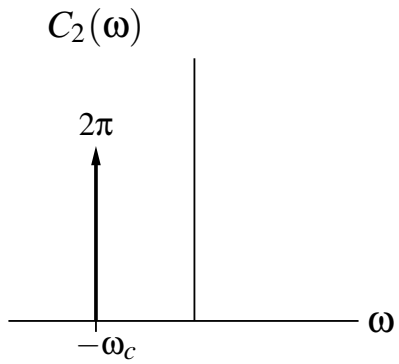
- Clearly, $\hat{x}(t) = e^{j\omega_c t} e^{-j\omega_c t} x(t) = x(t)$.

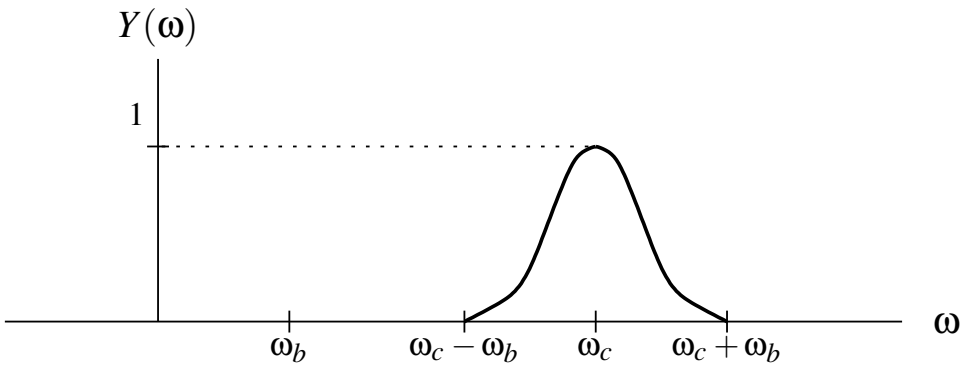
Trivial Amplitude Modulation (AM) System: Example



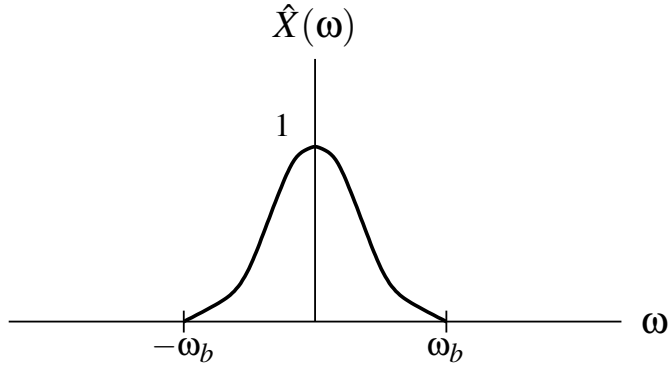
Transmitter Input





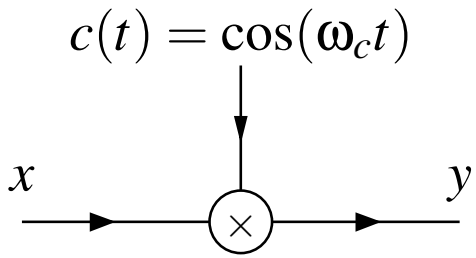


Transmitter Output

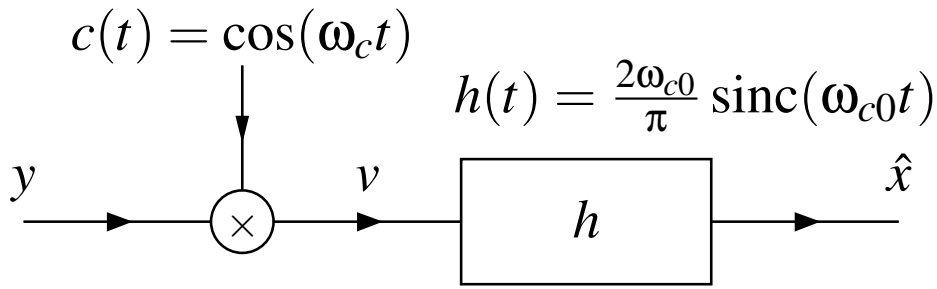


Receiver Output

Double-Sideband Suppressed-Carrier (DSB-SC) AM



Transmitter



Receiver

- Let $X = \mathcal{F}x$, $Y = \mathcal{F}y$, and $\hat{X} = \mathcal{F}\hat{x}$.
- Suppose that $X(\omega) = 0$ for all $\omega \notin [-\omega_b, \omega_b]$.
- The transmitter is characterized by

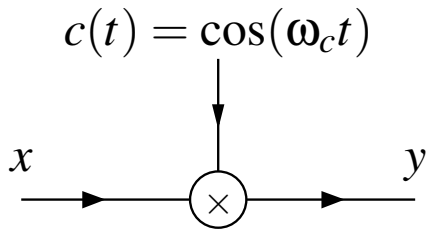
$$Y(\omega) = \tfrac{1}{2}\left[X(\omega + \omega_c) + X(\omega - \omega_c)\right].$$

- The receiver is characterized by

$$\hat{X}(\omega) = \left[Y(\omega + \omega_c) + Y(\omega - \omega_c)\right] \mathrm{rect}\left(\frac{\omega}{2\omega_{c_0}}\right).$$

- If $\omega_b < \omega_{c_0} < 2\omega_c - \omega_b$, we have $\hat{X}(\omega) = X(\omega)$ (implying $\hat{x}(t) = x(t)$).

DSB-SC AM: Transmitter

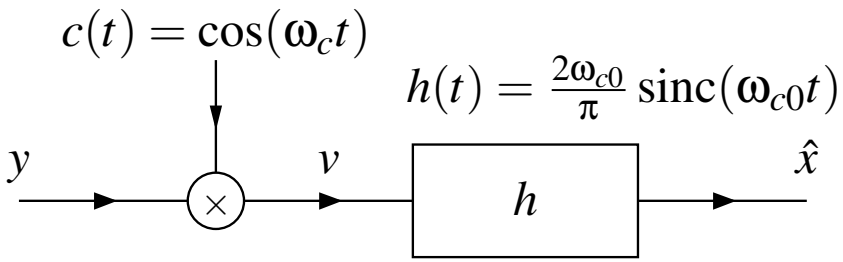


$$y(t) = \cos(\omega_c t)x(t)$$

$$X = \mathcal{F}x, \quad Y = \mathcal{F}y$$

$$\begin{aligned}
Y(\omega) &= \mathcal{F}\{\cos(\omega_c t)x(t)\}(\omega) \\
&= \mathcal{F}\left\{\tfrac{1}{2}\left(e^{j\omega_c t} + e^{-j\omega_c t}\right)x(t)\right\}(\omega) \\
&= \tfrac{1}{2}\left[\mathcal{F}\{e^{j\omega_c t}x(t)\}(\omega) + \mathcal{F}\{e^{-j\omega_c t}x(t)\}(\omega)\right] \\
&= \tfrac{1}{2}\left[X(\omega - \omega_c) + X(\omega + \omega_c)\right]
\end{aligned}$$

DSB-SC AM: Receiver



$$v(t) = \cos(\omega_c t) y(t), \quad h(t) = \frac{2\omega_{c0}}{\pi}\,\mathrm{sinc}(\omega_{c0} t), \quad \hat{x}(t) = v * h(t)$$

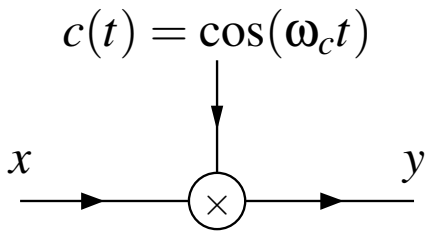
$$Y = \mathcal{F}y, \quad V = \mathcal{F}v, \quad H = \mathcal{F}h, \quad \hat{X} = \mathcal{F}\hat{x}$$

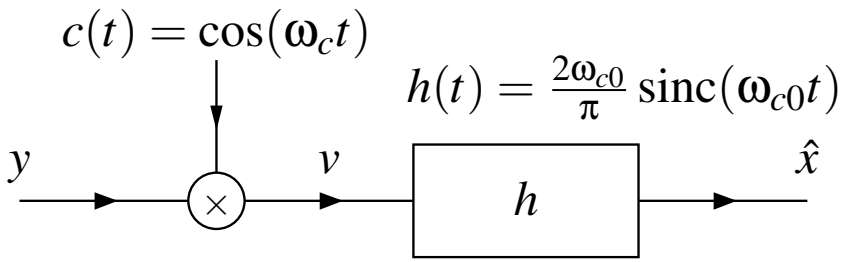
$$\begin{aligned} V(\omega) &= \mathcal{F}\{\cos(\omega_c t) y(t)\}(\omega) \\ &= \mathcal{F}\left\{\tfrac{1}{2}\left(e^{j\omega_c t} + e^{-j\omega_c t}\right) y(t)\right\}(\omega) \\ &= \tfrac{1}{2}\left[\mathcal{F}\left\{e^{j\omega_c t} y(t)\right\}(\omega) + \mathcal{F}\left\{e^{-j\omega_c t} y(t)\right\}(\omega)\right] \\ &= \tfrac{1}{2}[Y(\omega - \omega_c) + Y(\omega + \omega_c)] \end{aligned}$$

$$\begin{aligned} H(\omega) &= \mathcal{F}\left\{\frac{2\omega_{c0}}{\pi}\,\mathrm{sinc}(\omega_{c0} t)\right\}(\omega) \\ &= 2\,\mathrm{rect}\left(\frac{\omega}{2\omega_{c0}}\right) \end{aligned}$$

$$\hat{X}(\omega) = H(\omega) V(\omega)$$

DSB-SC AM: Complete System



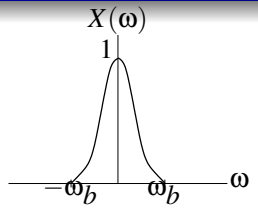


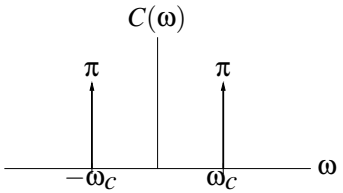
$$Y(\omega) = \tfrac{1}{2}\left[X(\omega-\omega_c)+X(\omega+\omega_c)\right]$$

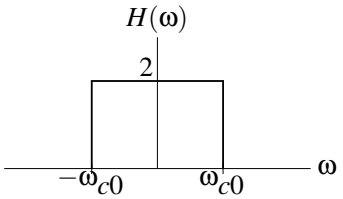
$$\begin{aligned}
V(\omega) &= \tfrac{1}{2}\left[Y(\omega-\omega_c)+Y(\omega+\omega_c)\right] \\
&= \tfrac{1}{2}\left[\tfrac{1}{2}\left[X(\left[\omega-\omega_c\right]-\omega_c)+X(\left[\omega-\omega_c\right]+\omega_c)\right]+\right. \\
&\qquad \left.\tfrac{1}{2}\left[X(\left[\omega+\omega_c\right]-\omega_c)+X(\left[\omega+\omega_c\right]+\omega_c)\right]\right] \\
&= \tfrac{1}{2}X(\omega)+\tfrac{1}{4}X(\omega-2\omega_c)+\tfrac{1}{4}X(\omega+2\omega_c)
\end{aligned}$$

$$\begin{aligned}
\hat{X}(\omega) &= H(\omega)V(\omega) \\
&= H(\omega)\left[\tfrac{1}{2}X(\omega)+\tfrac{1}{4}X(\omega-2\omega_c)+\tfrac{1}{4}X(\omega+2\omega_c)\right] \\
&= \tfrac{1}{2}H(\omega)X(\omega)+\tfrac{1}{4}H(\omega)X(\omega-2\omega_c)+\tfrac{1}{4}H(\omega)X(\omega+2\omega_c) \\
&= \tfrac{1}{2}\left[2X(\omega)\right]+\tfrac{1}{4}(0)+\tfrac{1}{4}(0) \\
&= X(\omega)
\end{aligned}$$

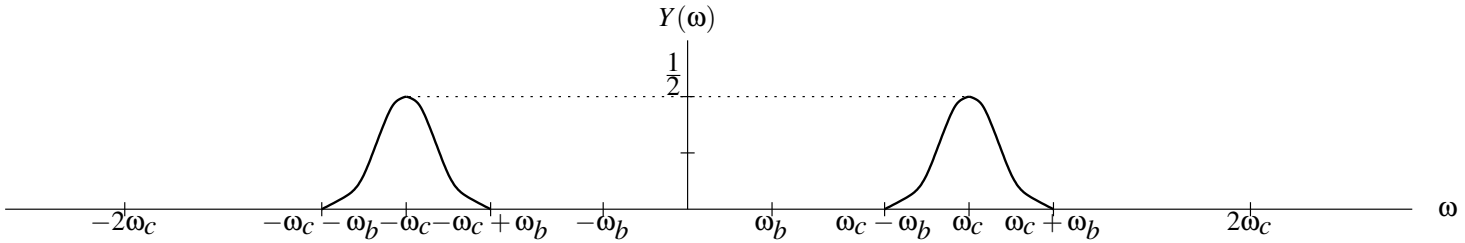
DSB-SC AM: Example



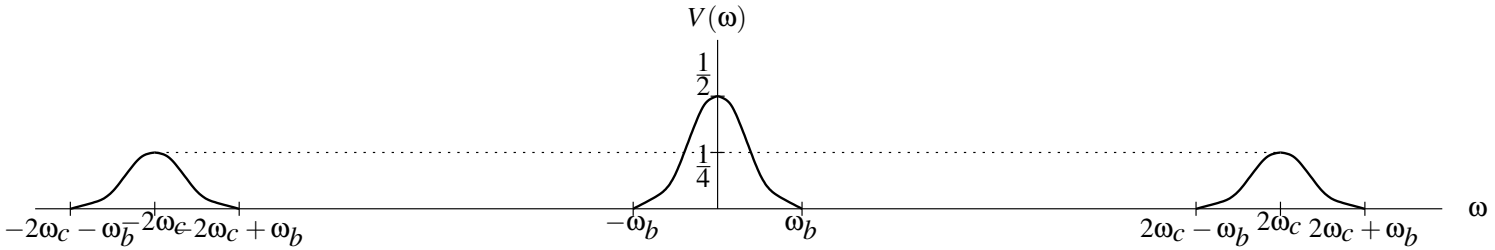


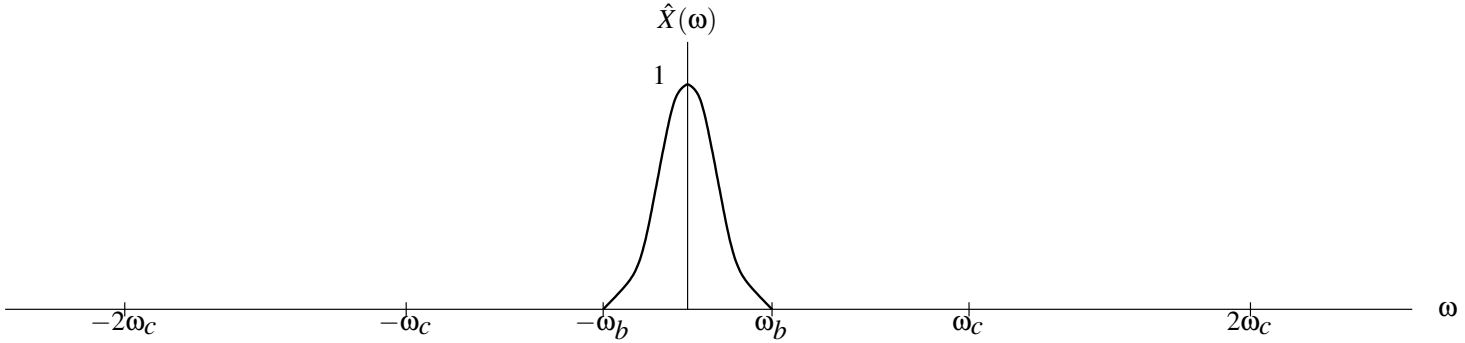


Transmitter Input



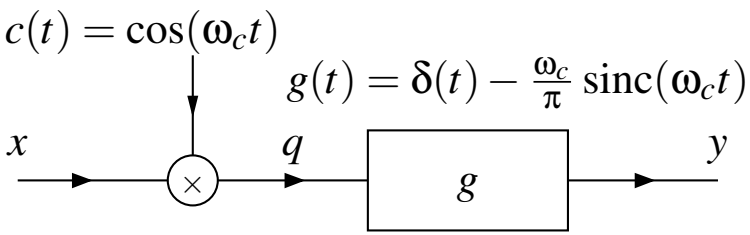
Transmitter Output



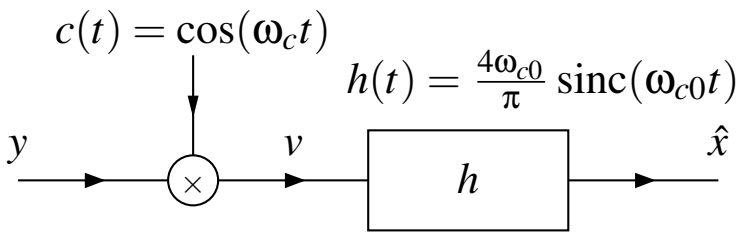


Receiver Output

Single-Sideband Suppressed-Carrier (SSB-SC) AM



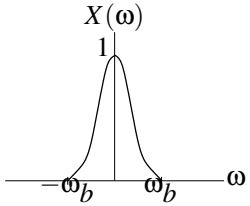
Transmitter

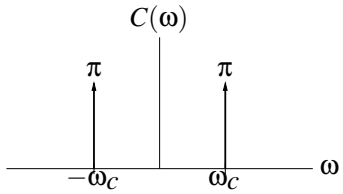


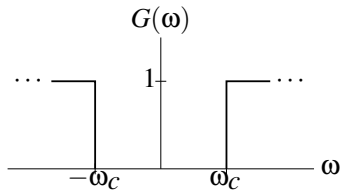
Receiver

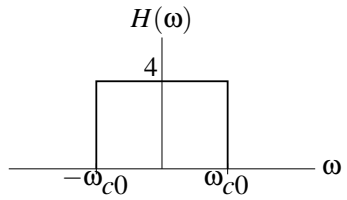
- The basic analysis of the SSB-SC AM system is similar to the DSB-SC AM system.
- SSB-SC AM requires half as much bandwidth for the transmitted signal as DSB-SC AM.

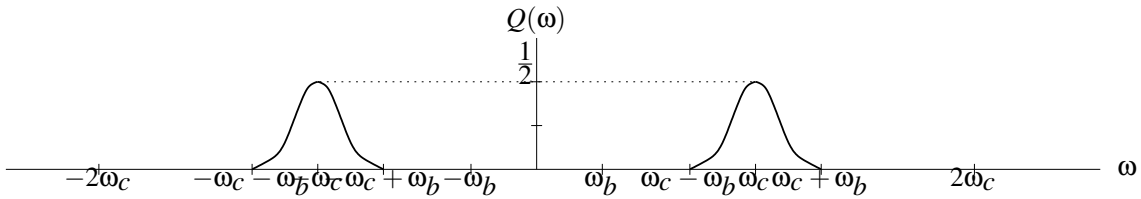
SSB-SC AM: Example

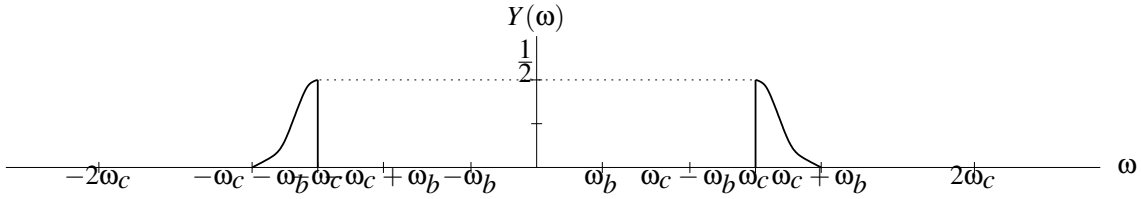


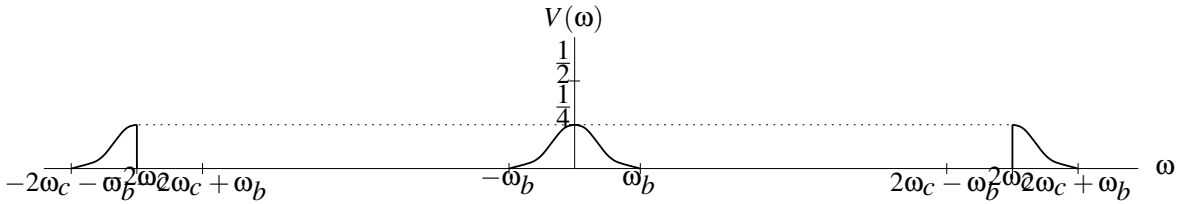


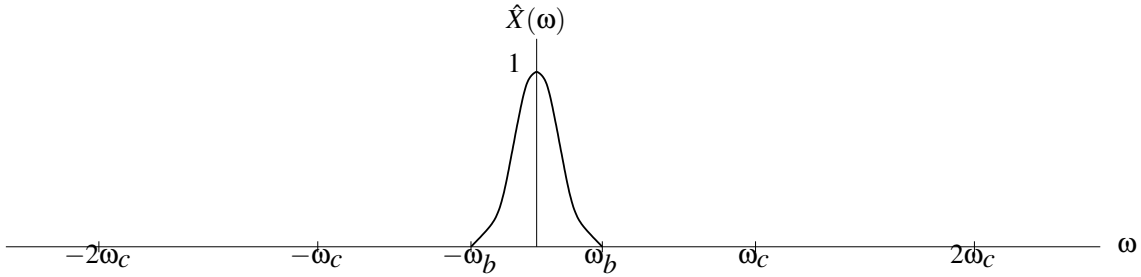










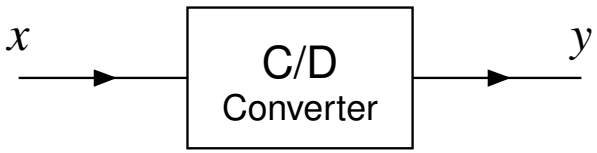


Section 6.11

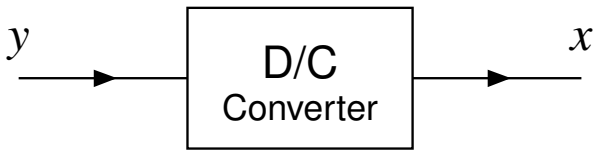
Application: Sampling and Interpolation

Sampling and Interpolation

- Often, we want to be able to transform a continuous-time signal (i.e., a function) into a discrete-time signal (i.e., a sequence) and vice versa.
- This is accomplished through processes known as ***sampling*** and ***interpolation***.
- **Sampling**, which is performed by a **continuous-time to discrete-time (C/D) converter** shown below, transforms a function x to a sequence y.



- **Interpolation**, which is performed by a **discrete-time to continuous-time (D/C) converter** shown below, transforms a sequence y to a function x.



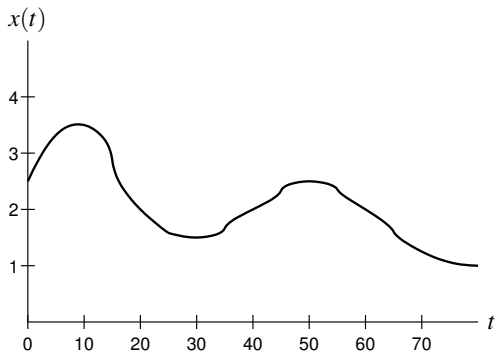
- Note that, unless very special conditions are met, the sampling process loses information (i.e., is ***not invertible***).

Periodic Sampling

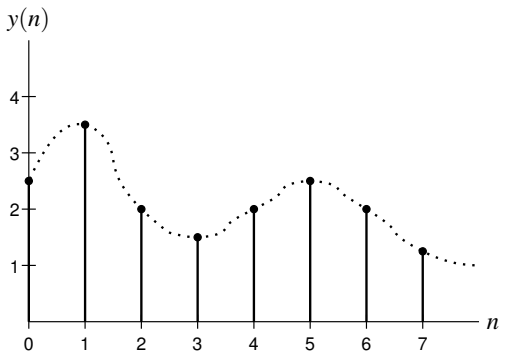
- Although sampling can be performed in many different ways, the most commonly used scheme is **periodic sampling**.
- With this scheme, a sequence y of samples is obtained from a function x according to the relation

$$y(n) = x(Tn) \quad \text{for all integer } n,$$

  where T is a (strictly) positive real constant.
- As a matter of terminology, we refer to T as the **sampling period**, and $\omega_s = \frac{2\pi}{T}$ as the (angular) **sampling frequency**.
- An example of periodic sampling is shown below, where the function x has been sampled with ***sampling period*** $T = 10$, yielding the sequence y.



Function to Be Sampled



Sequence Produced by Sampling

Invertibility of Sampling

- Unless constraints are placed on the functions being sampled, the sampling process is ***not invertible***.
- In other words, in the absence of any constraints, a function cannot be uniquely determined from a sequence of its equally-spaced samples.
- Consider, for example, the functions x_1 and x_2 given by

$$x_1(t) = 0 \quad \text{and} \quad x_2(t) = \sin(2\pi t).$$

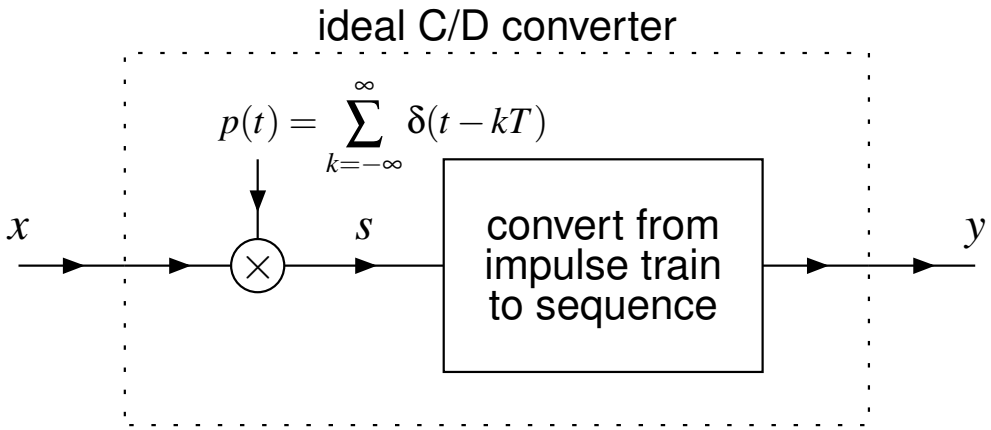
- Sampling x_1 and x_2 with the sampling period $T = 1$ yields the respective sequences

$$y_1(n) = x_1(Tn) = x_1(n) = 0 \quad \text{and}$$
$$y_2(n) = x_2(Tn) = \sin(2\pi n) = 0.$$

- So, although x_1 and x_2 are ***distinct***, y_1 and y_2 are ***identical***.
- Given the sequence y where $y = y_1 = y_2$, it is impossible to determine which function was sampled to produce y.
- Only by imposing a carefully chosen set of constraints on the functions being sampled can we ensure that a function can be exactly recovered from only its samples.

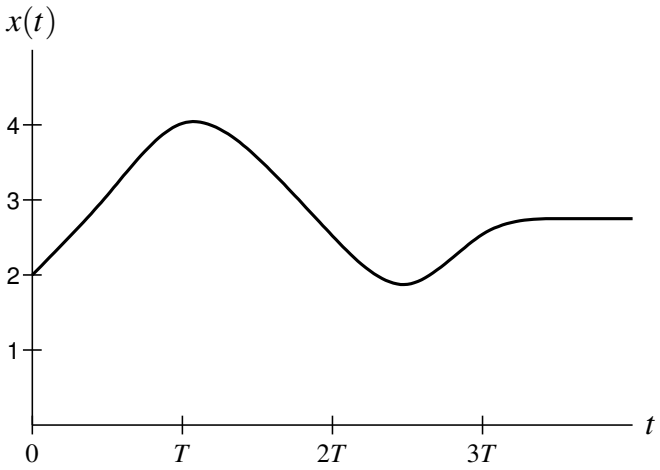
Model of Sampling

- An **impulse train** is a function of the form $v(t) = \sum_{k=-\infty}^{\infty} c_k \delta(t - kT)$, where c_k and T are real constants.
- For the purposes of analysis, sampling with sampling period T and frequency $\omega_s = \frac{2\pi}{T}$ can be modelled as shown below.

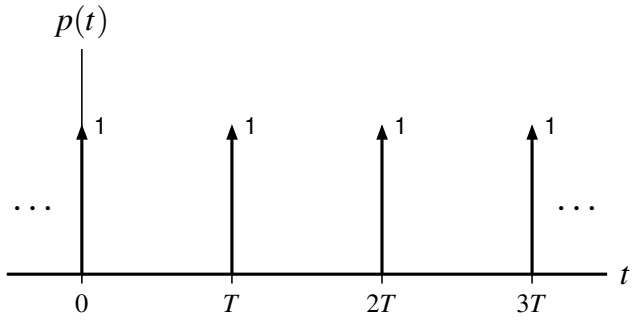


- The sampling of a function x to produce a sequence y consists of the following two steps (in order):
  1. Multiply the function x to be sampled by a periodic impulse train p, yielding the impulse train $s(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)$.
  2. Convert the impulse train s to a sequence y by forming y from the weights of successive impulses in s so that $y(n) = x(nT)$.

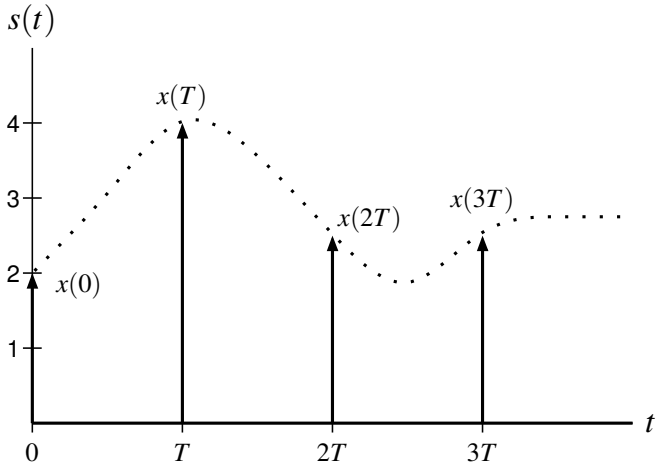
Model of Sampling: Various Signals



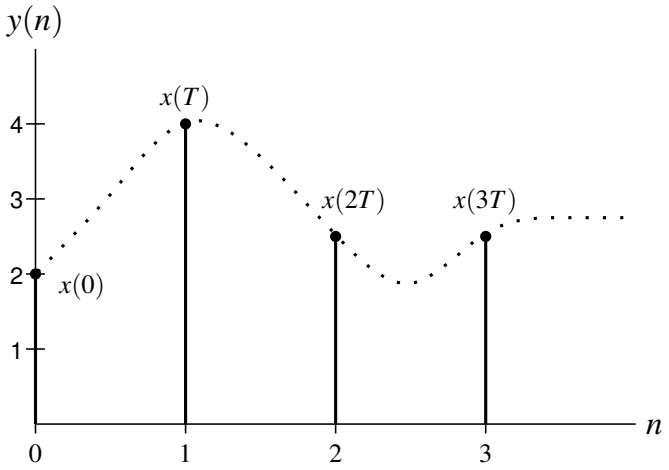
Input Function



Periodic Impulse Train

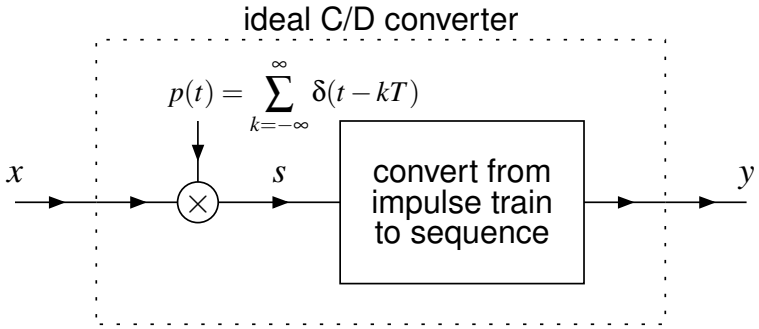


Impulse-Sampled Function
(Continuous-Time)



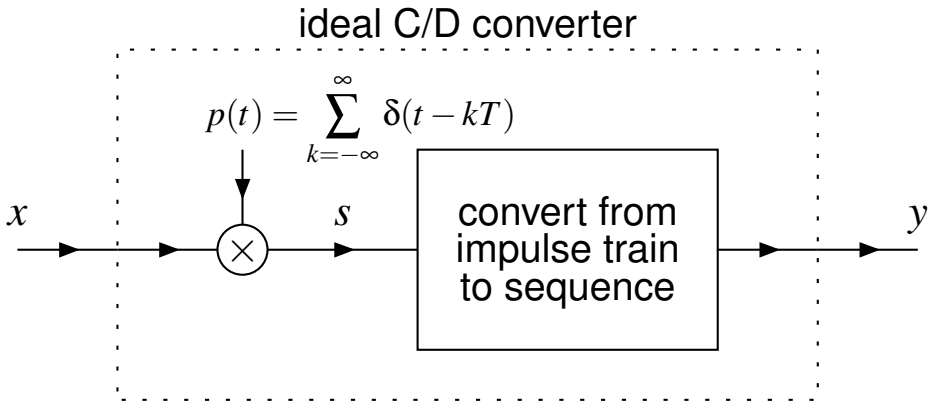
Output Sequence *(Discrete-Time)*

Model of Sampling: Invertibility of Sampling Revisited



- Since sampling is not invertible and our model of sampling consists of only two steps, at least one of these two steps must not be invertible.
- Recall the two steps in our model of sampling are as follows (in order):
  1. $x \longrightarrow s(t) = x(t)p(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)$; and
  2. $s(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT) \longrightarrow y(n) = x(nT)$.
- Step 1 cannot be undone (unless we somehow restrict which functions x can be sampled).
- Step 2 is always invertible.
- Therefore, the fact that sampling is not invertible is entirely due to step 1.

Model of Sampling: Characterization



- In the time domain, the impulse-sampled function s is given by

$$s(t) = x(t)p(t) \quad \text{where} \quad p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT).$$

- In the Fourier domain, the preceding equation becomes

$$S(\omega) = \frac{\omega_s}{2\pi} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s) \quad (\text{where } \omega_s = \tfrac{2\pi}{T}).$$

- Thus, the spectrum of the impulse-sampled function s is a scaled sum of an infinite number of ***shifted copies*** of the spectrum of the original function x.

Sampling: Fourier Series for a Periodic Impulse Train

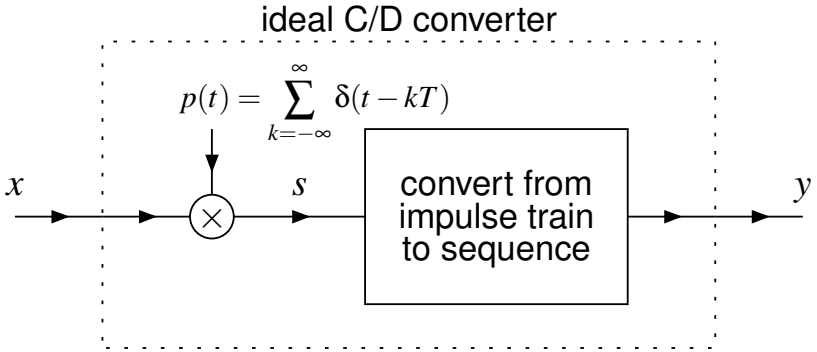
$$p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT), \quad \omega_s = \frac{2\pi}{T}$$

$$p(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_s t}$$

$$\begin{aligned} c_k &= \frac{1}{T} \int_{-T/2}^{T/2} p(t) e^{-jk\omega_s t} dt \\ &= \frac{1}{T} \int_{-T/2}^{T/2} \delta(t) e^{-jk\omega_s t} dt \\ &= \frac{1}{T} \int_{-\infty}^{\infty} \delta(t) e^{-jk\omega_s t} dt \\ &= \frac{1}{T} \\ &= \frac{\omega_s}{2\pi} \end{aligned}$$

$$p(t) = \frac{\omega_s}{2\pi} \sum_{k=-\infty}^{\infty} e^{jk\omega_s t}$$

Sampling: Multiplication by a Periodic Impulse Train



$$s(t) = p(t)x(t), \quad p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT), \quad \omega_s = \frac{2\pi}{T}$$

$$p(t) = \frac{\omega_s}{2\pi} \sum_{k=-\infty}^{\infty} e^{jk\omega_s t}$$

$$s(t) = \frac{\omega_s}{2\pi} \sum_{k=-\infty}^{\infty} e^{jk\omega_s t} x(t)$$

$$X = \mathcal{F}x, \quad S = \mathcal{F}s$$

$$S(\omega) = \frac{\omega_s}{2\pi} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)$$

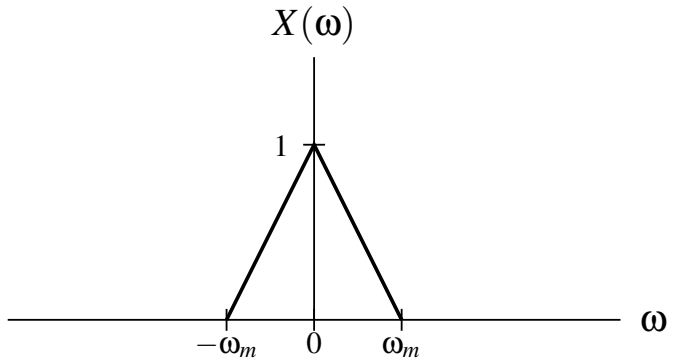
Model of Sampling: Aliasing

- Consider frequency spectrum S of the impulse-sampled function s given by

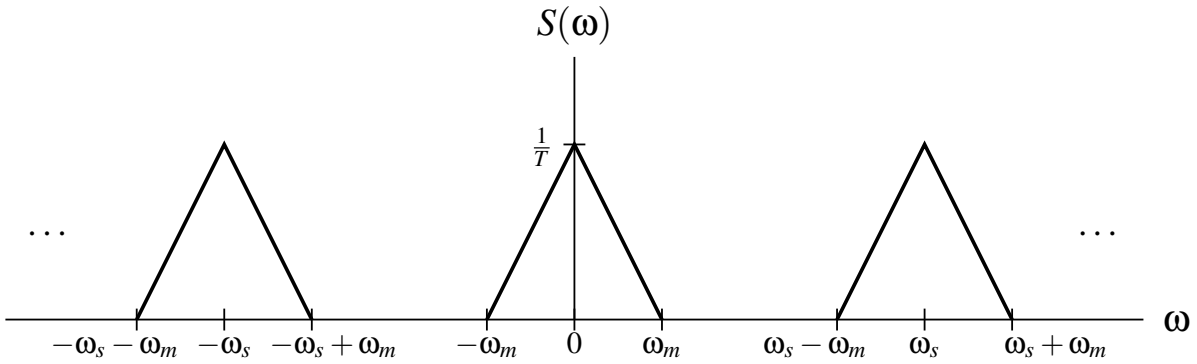
$$S(\omega) = \frac{\omega_s}{2\pi} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s).$$

- The function S is a scaled sum of an infinite number of ***shifted copies*** of X.
- Two distinct behaviors can result in this summation, depending on ω_s and the bandwidth of x.
- In particular, the nonzero portions of the different shifted copies of X can either:
  1. overlap; or
  2. not overlap.
- In the case where overlap occurs, the various shifted copies of X add together in such a way that the original shape of X is lost. This phenomenon is known as **aliasing**.
- When aliasing occurs, the original function x cannot be recovered from its samples in y.

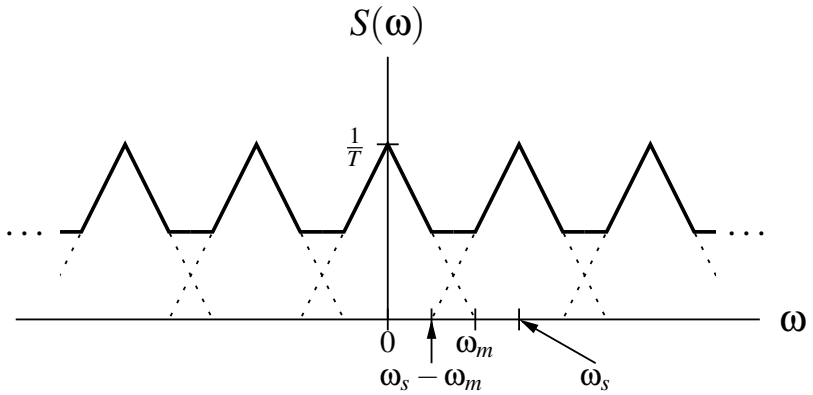
Model of Sampling: Aliasing (Continued)



Spectrum of Input Function (Bandwidth ω_m)



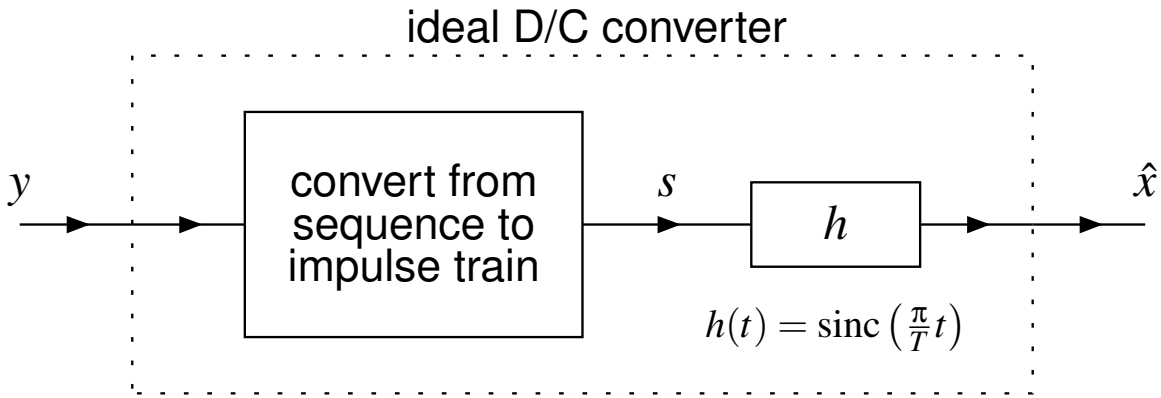
Spectrum of Impulse-Sampled Function: No Aliasing Case ($\omega_s > 2\omega_m$)



Spectrum of Impulse-Sampled Function: Aliasing Case ($\omega_s \leq 2\omega_m$)

Model of Interpolation

- For the purposes of analysis, interpolation can be modelled as shown below.



- The reconstruction of a function x from its sequence y of samples (i.e., bandlimited interpolation) consists of the following two steps (in order):
  1. Convert the sequence y to the impulse train s by using the samples in y as the weights of successive impulses in s so that $s(t) = \sum_{n=-\infty}^{\infty} y(n)\delta(t - Tn)$.
  2. Apply the lowpass filter with impulse response h to s to produce $\hat{x}$ so that $\hat{x}(t) = s * h(t) = \sum_{n=-\infty}^{\infty} y(n)\,\text{sinc}\left[\frac{\pi}{T}(t - Tn)\right]$.
- The lowpass filter is used to eliminate the extra copies of the originally-sampled function's spectrum present in the spectrum of s.

Sampling Theorem

- **Sampling Theorem.** Let x be a function with Fourier transform X, and suppose that $|X(\omega)| = 0$ for all ω satisfying $|\omega| > \omega_M$ (i.e., x is bandlimited to frequencies $[-\omega_M, \omega_M]$). Then, x is uniquely determined by its samples $y(n) = x(Tn)$ for all integer n, if

$$\omega_s > 2\omega_M,$$

where $\omega_s = \frac{2\pi}{T}$. The preceding inequality is known as the **Nyquist condition**. If this condition is satisfied, we have that

$$x(t) = \sum_{n=-\infty}^{\infty} y(n) \operatorname{sinc}\left[\frac{\pi}{T}(t - Tn)\right],$$

or equivalently (i.e., rewritten in terms of ω_s instead of T),

$$x(t) = \sum_{n=-\infty}^{\infty} y(n) \operatorname{sinc}\left(\frac{\omega_s}{2}t - \pi n\right).$$

- We call $\frac{\omega_s}{2}$ the **Nyquist frequency** and $2\omega_M$ the **Nyquist rate**.

Part 7

Laplace Transform (LT)

Motivation Behind the Laplace Transform

- Another important mathematical tool in the study of signals and systems is known as the Laplace transform.
- The Laplace transform can be viewed as a ***generalization of the (classical) Fourier transform***.
- Due to its more general nature, the Laplace transform has a number of ***advantages*** over the (classical) Fourier transform.
- First, the Laplace transform representation ***exists for some functions that do not have a Fourier transform representation***. So, we can handle some functions with the Laplace transform that cannot be handled with the Fourier transform.
- Second, since the Laplace transform is a more general tool, it can provide ***additional insights*** beyond those facilitated by the Fourier transform.

Motivation Behind the Laplace Transform (Continued)

- Earlier, we saw that complex exponentials are eigenfunctions of LTI systems.
- In particular, for a LTI system $\mathcal{H}$ with impulse response h, we have that

$$\mathcal{H}\{e^{st}\}(t) = H(s)e^{st} \quad \text{where} \quad H(s) = \int_{-\infty}^{\infty} h(t)e^{-st}dt.$$

- Previously, we referred to H as the system function.
- As it turns out, H is the Laplace transform of h.
- Since the Laplace transform has already appeared earlier in the context of LTI systems, it is clearly a useful tool.
- Furthermore, as we will see, the Laplace transform has many additional uses.

Section 7.1

Laplace Transform

(Bilateral) Laplace Transform

- The (bilateral) **Laplace transform** of the function x, denoted $\mathcal{L}x$ or X, is defined as

$$\mathcal{L}x(s) = X(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt.$$

- The **inverse Laplace transform** of X, denoted $\mathcal{L}^{-1}X$ or x, is then given by

$$\mathcal{L}^{-1}X(t) = x(t) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st}ds,$$

  where $\mathrm{Re}(s) = \sigma$ is in the ROC of X. (Note that this is a ***contour integration***, since s is complex.)
- We refer to x and X as a **Laplace transform pair** and denote this relationship as

$$x(t) \overset{\text{LT}}{\longleftrightarrow} X(s).$$

- In practice, we do not usually compute the inverse Laplace transform by directly using the formula from above. Instead, we resort to other means (to be discussed later).

Bilateral and Unilateral Laplace Transforms

- Two different versions of the Laplace transform are commonly used:
  1. the ***bilateral*** (or ***two-sided***) Laplace transform; and
  2. the ***unilateral*** (or ***one-sided***) Laplace transform.
- The unilateral Laplace transform is most frequently used to solve systems of linear differential equations with nonzero initial conditions.
- As it turns out, the only difference between the definitions of the bilateral and unilateral Laplace transforms is in the ***lower limit of integration***.
- In the bilateral case, the lower limit is $-\infty$, whereas in the unilateral case, the lower limit is 0 (i.e., $\int_{-\infty}^{\infty} x(t)e^{-st}dt$ versus $\int_{0^-}^{\infty} x(t)e^{-st}dt$).
- For the most part, we will focus our attention primarily on the bilateral Laplace transform.
- We will, however, briefly introduce the unilateral Laplace transform as a tool for solving differential equations.
- Unless otherwise noted, all subsequent references to the Laplace transform should be understood to mean ***bilateral*** Laplace transform.

Remarks on Operator Notation

- For a function x, the Laplace transform of x is denoted using operator notation as $\mathcal{L}x$.
- The Laplace transform of x evaluated at s is denoted $\mathcal{L}x(s)$.
- Note that $\mathcal{L}x$ is a function, whereas $\mathcal{L}x(s)$ is a number.
- Similarly, for a function X, the inverse Laplace transform of X is denoted using operator notation as $\mathcal{L}^{-1}X$.
- The inverse Laplace transform of X evaluated at t is denoted $\mathcal{L}^{-1}X(t)$.
- Note that $\mathcal{L}^{-1}X$ is a function, whereas $\mathcal{L}^{-1}X(t)$ is a number.
- With the above said, engineers often abuse notation, and use expressions like those above to mean things different from their proper meanings.
- Since such notational abuse can lead to problems, it is strongly recommended that one refrain from doing this.

Remarks on Dot Notation

- Often, we would like to write an expression for the Laplace transform of a function without explicitly naming the function.
- For example, consider writing an expression for the Laplace transform of the function $v(t) = x(5t-3)$ but without using the name "v".
- It would be incorrect to write "$\mathcal{L}x(5t-3)$" as this is the function $\mathcal{L}x$ evaluated at $5t-3$, which is not the meaning that we wish to convey.
- Also, strictly speaking, it would be incorrect to write "$\mathcal{L}\{x(5t-3)\}$" as the operand of the Laplace transform operator must be a function, and $x(5t-3)$ is a number (i.e., the function x evaluated at $5t-3$).
- Using dot notation, we can write the following strictly-correct expression for the desired Laplace transform: $\mathcal{L}\{x(5\cdot-3)\}$.
- In many cases, however, it is probably advisable to avoid employing anonymous (i.e., unnamed) functions, as their use tends to be more error prone in some contexts.

Remarks on Notational Conventions

- Since dot notation is less frequently used by engineers, the author has elected to minimize its use herein.
- To avoid ambiguous notation, the following conventions are followed:
  1. in the expression for the operand of a Laplace transform operator, the ***independent variable is assumed to be the variable named "t"*** unless otherwise indicated (i.e., in terms of dot notation, each "t" is treated as if it were a "$\cdot$")
  2. in the expression for the operand of the inverse Laplace transform operator, the ***independent variable is assumed to be the variable named "s"*** unless otherwise indicated (i.e., in terms of dot notation, each "s" is treated as if it were a "$\cdot$").
- For example, with these conventions:
  - "$\mathcal{L}\{(t-\tau)u(t-\tau)\}$" denotes the function that is the Laplace transform of the function $v(t) = (t-\tau)u(t-\tau)$ (not the Laplace transform of the function $v(\tau) = (t-\tau)u(t-\tau)$).
  - "$\mathcal{L}^{-1}\{\frac{1}{s^2-\lambda}\}$" denotes the function that is the inverse Laplace transform of the function $V(s) = \{\frac{1}{s^2-\lambda}\}$ (not the inverse Laplace transform of the function $V(\lambda) = \{\frac{1}{s^2-\lambda}\}$).

Relationship Between Laplace and Fourier Transforms

- Let X and X_{F} denote the Laplace and (CT) Fourier transforms of x, respectively.
- The function X evaluated at $j\omega$ (where ω is real) yields $X_{\mathrm{F}}(\omega)$. That is,

$$X(j\omega) = X_{\mathrm{F}}(\omega).$$

- Due to the preceding relationship, the Fourier transform of x is sometimes written as $X(j\omega)$.
- The function X evaluated at an arbitrary complex value $s = \sigma + j\omega$ (where $\sigma = \mathrm{Re}(s)$ and $\omega = \mathrm{Im}(s)$) can also be expressed in terms of a Fourier transform involving x. In particular, we have

$$X(\sigma + j\omega) = X'_{\mathrm{F}}(\omega),$$

  where X'_{F} is the (CT) Fourier transform of $x'(t) = e^{-\sigma t}x(t)$.
- So, in general, the Laplace transform of x is the Fourier transform of an exponentially-weighted version of x.
- Due to this weighting, the Laplace transform of a function may exist when the Fourier transform of the same function does not.

Laplace Transform Examples

THIS SLIDE IS INTENTIONALLY LEFT BLANK.

Section 7.2

Region of Convergence (ROC)

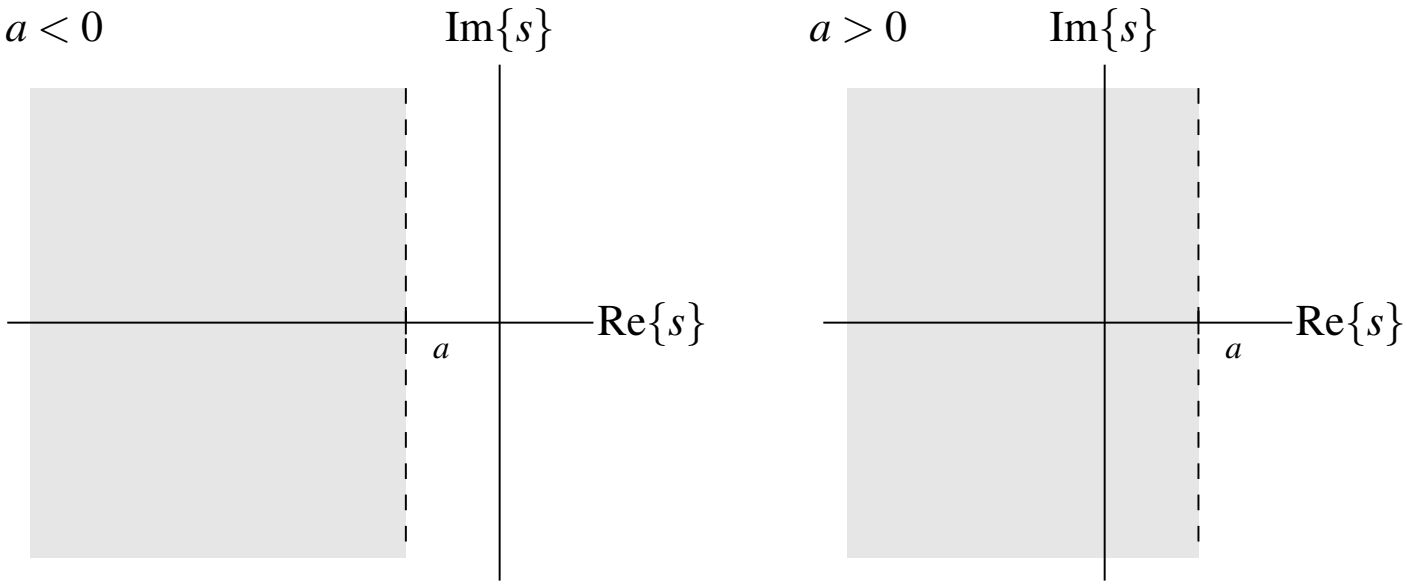
Left-Half Plane (LHP)

- The set R of all complex numbers s satisfying

$$\mathrm{Re}(s) < a$$

  for some real constant a is said to be a **left-half plane (LHP)**.

- Some examples of LHPs are shown below.



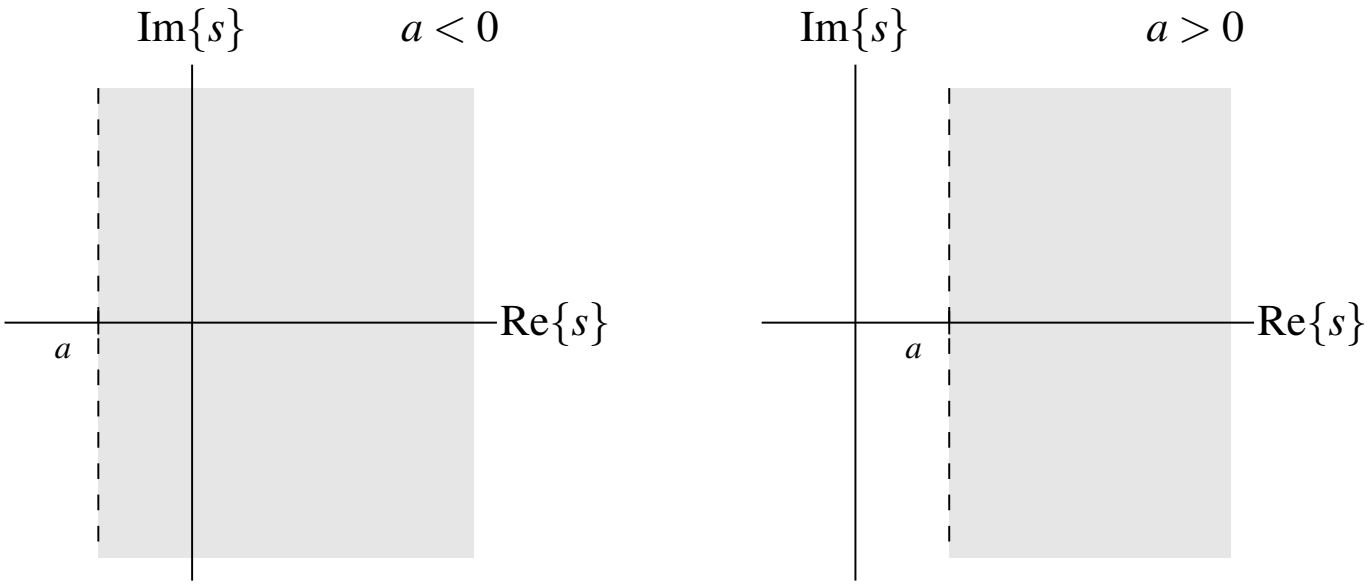
Right-Half Plane (RHP)

- The set R of all complex numbers s satisfying

$$\mathrm{Re}(s) > a$$

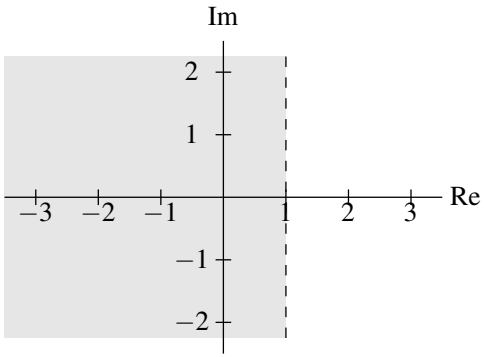
  for some real constant a is said to be a **right-half plane (RHP)**.

- Some examples of RHPs are shown below.

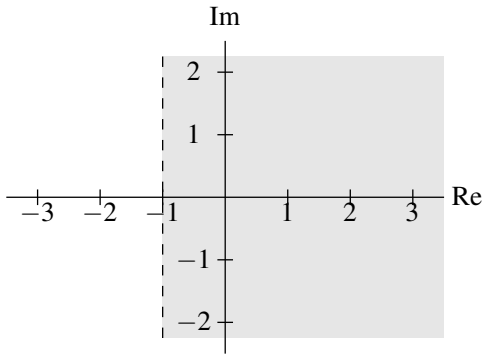


Intersection of Sets

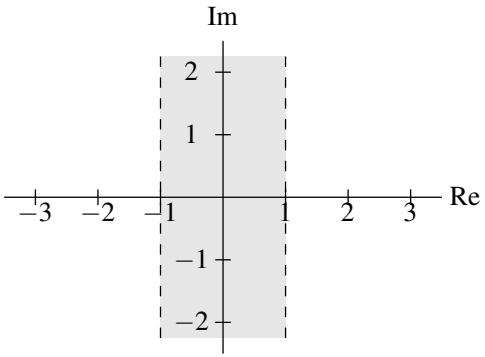
- For two sets A and B, the **intersection** of A and B, denoted $A \cap B$, is the set of all points that are in both A and B.
- An illustrative example of set intersection is shown below.



R_1



R_2



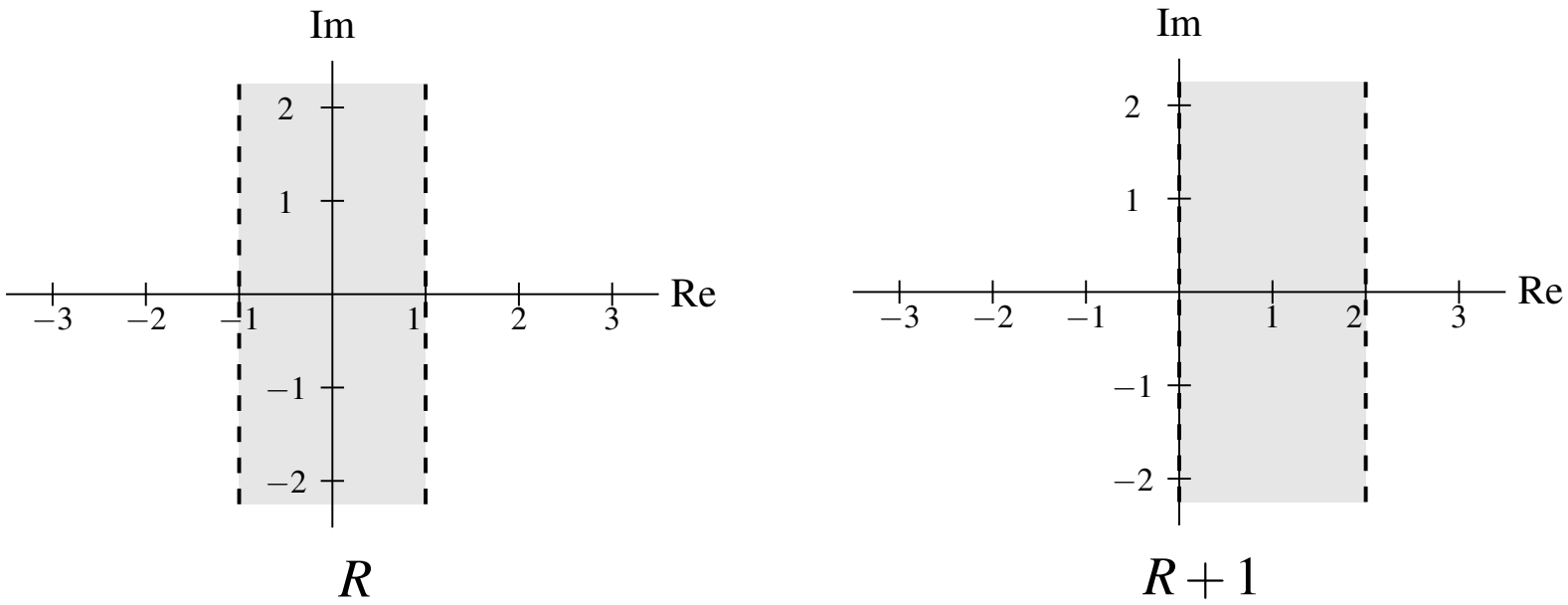
$R_1 \cap R_2$

Adding a Scalar to a Set

- For a set S and a scalar constant a, $S+a$ denotes the set given by

$$S+a=\{z+a:z\in S\}$$

  (i.e., $S+a$ is the set formed by adding a to each element of S).
- Effectively, adding a scalar to a set applies a translation (i.e., shift) to the region associated with the set.
- An illustrative example is given below.

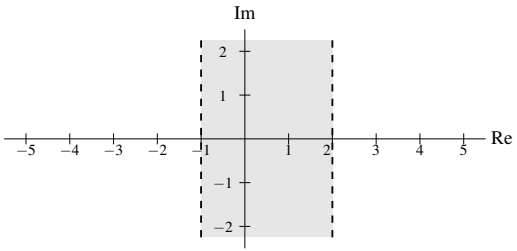


Multiplying a Set by a Scalar

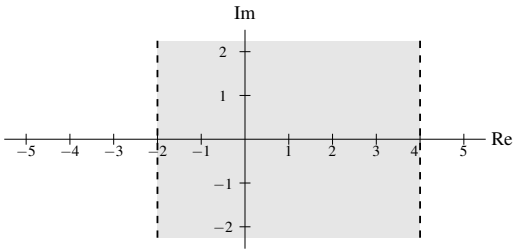
- For a set S and a scalar constant a, aS denotes the set given by

$$aS = \{az : z \in S\}$$

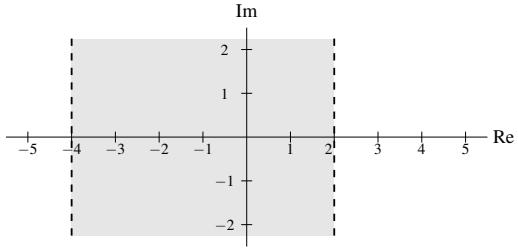
  (i..e, aS is the set formed by multiplying each element of S by a).
- Multiplying z by a affects z by: scaling by $|a|$ and rotating about the origin by $\arg a$.
- So, effectively, multiplying a set by a scalar applies a scaling and/or rotation to the region associated with the set.
- An illustrative example is given below.



R



$2R$



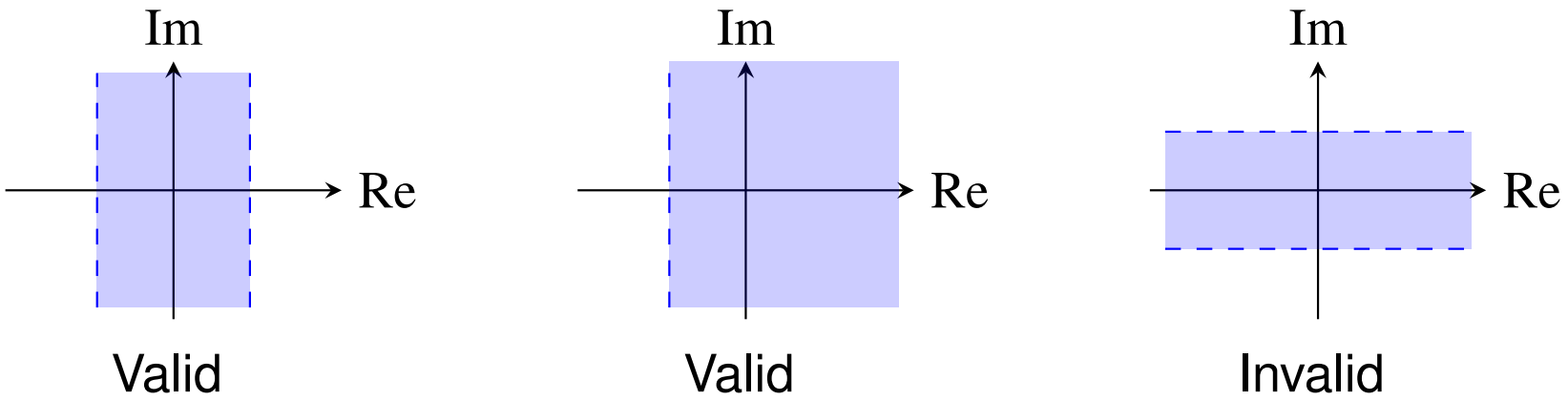
$-2R$

Region of Convergence (ROC)

- As we saw earlier, for a function x, the complete specification of its Laplace transform X requires not only an algebraic expression for X, but also the ROC associated with X.
- Two very different functions can have the same algebraic expressions for X.
- On the slides that follow, we will examine a number of key properties of the ROC of the Laplace transform.

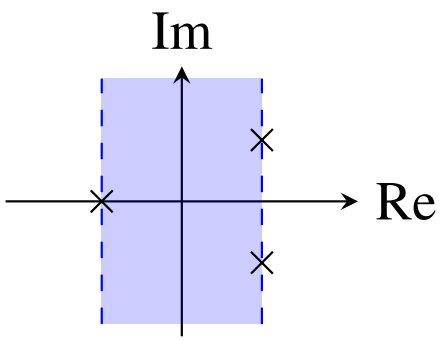
ROC Property 1: General Form

- The ROC of a Laplace transform consists of ***strips parallel to the imaginary axis*** in the complex plane.
- That is, if a point s_0 is in the ROC, then the vertical line through s_0 (i.e., $\mathrm{Re}(s) = \mathrm{Re}(s_0)$) is also in the ROC.
- Some examples of sets that would be either valid or invalid as ROCs are shown below.

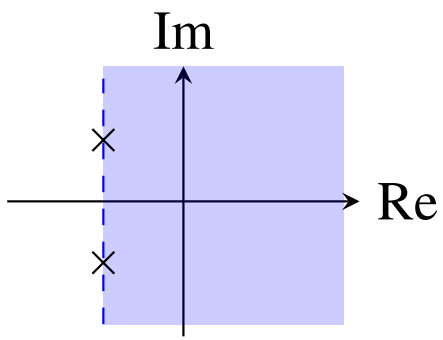


ROC Property 2: Rational Laplace Transforms

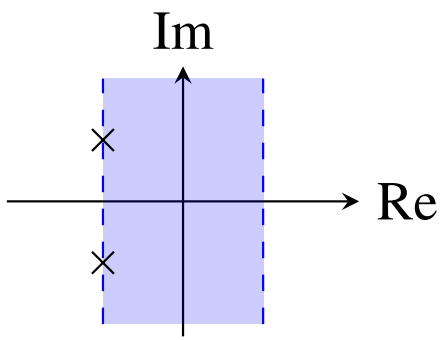
- If a Laplace transform X is a ***rational*** function, the ROC of X ***does not contain any poles*** and is ***bounded by poles or extends to infinity***.
- Some examples of sets that would be either valid or invalid as ROCs of rational Laplace transforms are shown below.



Valid



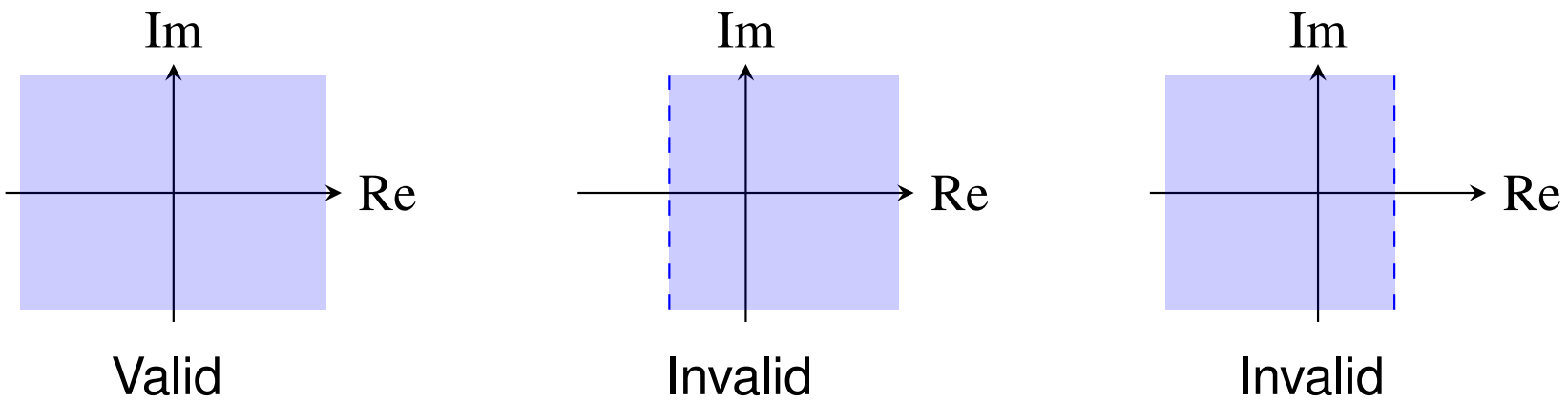
Valid



Invalid

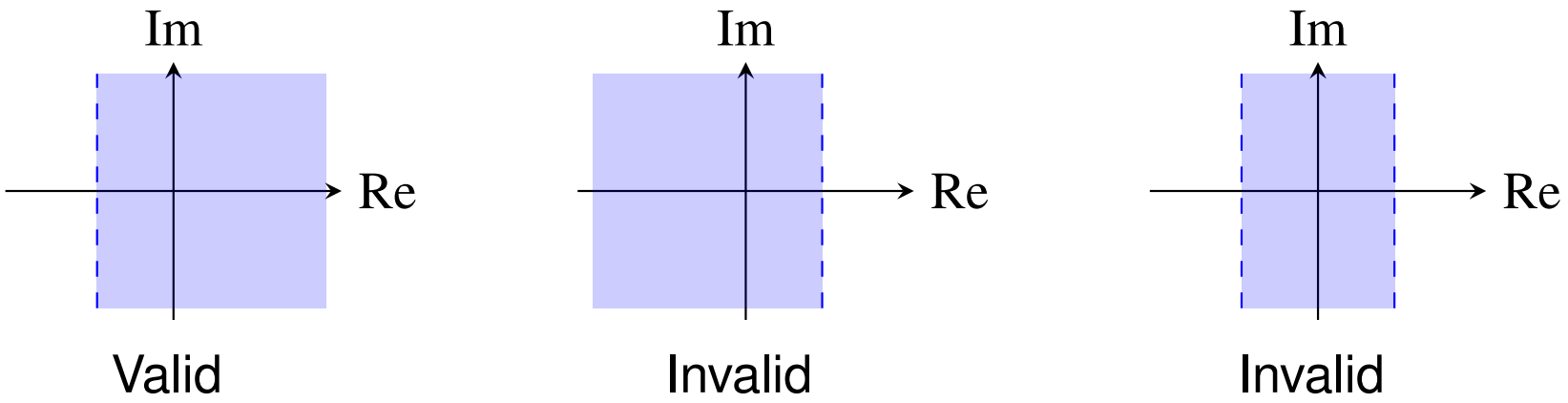
ROC Property 3: Finite-Duration Functions

- If a function x is ***finite duration*** and its Laplace transform X converges for at least one point, then X converges for ***all*** points in the complex plane (i.e., the ROC is the entire complex plane).
- Some examples of sets that would be either valid or invalid as ROCs for X, if x is finite duration, are shown below.



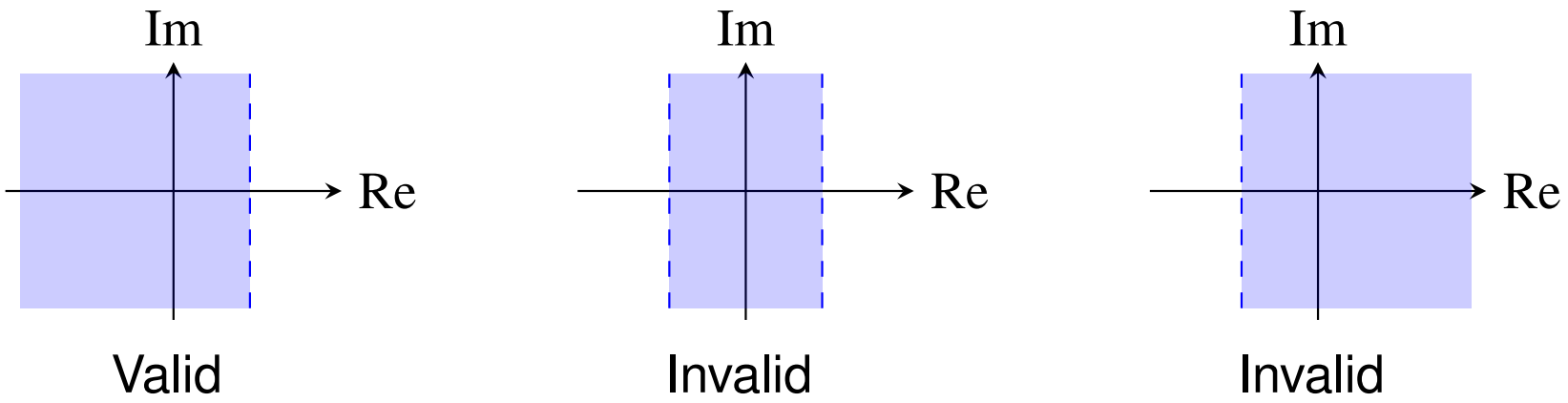
ROC Property 4: Right-Sided Functions

- If a function x is ***right sided*** and the (vertical) line $\mathrm{Re}(s) = \sigma_0$ is in the ROC of the Laplace transform $X = \mathcal{L}x$, then all values of s for which $\mathrm{Re}(s) > \sigma_0$ must also be in the ROC (i.e., the ROC includes a RHP containing $\mathrm{Re}(s) = \sigma_0$).
- Thus, if x is ***right sided but not left sided***, the ROC of X is a ***RHP***.
- Some examples of sets that would be either valid or invalid as ROCs for X, if x is right sided but not left sided, are shown below.



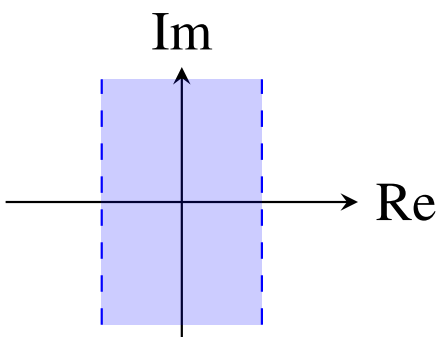
ROC Property 5: Left-Sided Functions

- If a function x is ***left sided*** and the (vertical) line $\mathrm{Re}(s) = \sigma_0$ is in the ROC of the Laplace transform $X = \mathcal{L}x$, then all values of s for which $\mathrm{Re}(s) < \sigma_0$ must also be in the ROC (i.e., the ROC includes a ***LHP*** containing $\mathrm{Re}(s) = \sigma_0$).
- Thus, if x is ***left sided but not right sided***, the ROC of X is a ***LHP***.
- Some examples of sets that would be either valid or invalid as ROCs for X, if x is left sided but not right sided, are shown below.

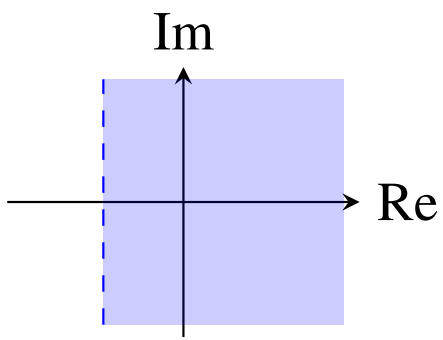


ROC Property 6: Two-Sided Functions

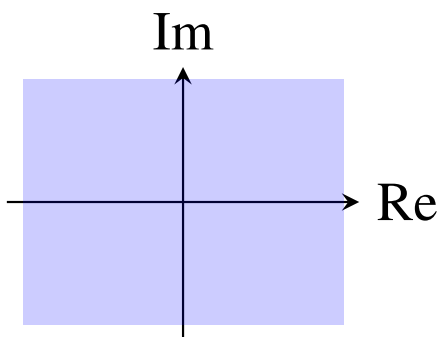
- If a function x is ***two sided*** and the (vertical) line $\mathrm{Re}(s) = \sigma_0$ is in the ROC of the Laplace transform $X = \mathcal{L}x$, then the ROC will consist of a ***strip*** in the complex plane that includes the line $\mathrm{Re}(s) = \sigma_0$.
- Some examples of sets that would be either valid or invalid as ROCs for X, if x is two sided, are shown below.



Valid



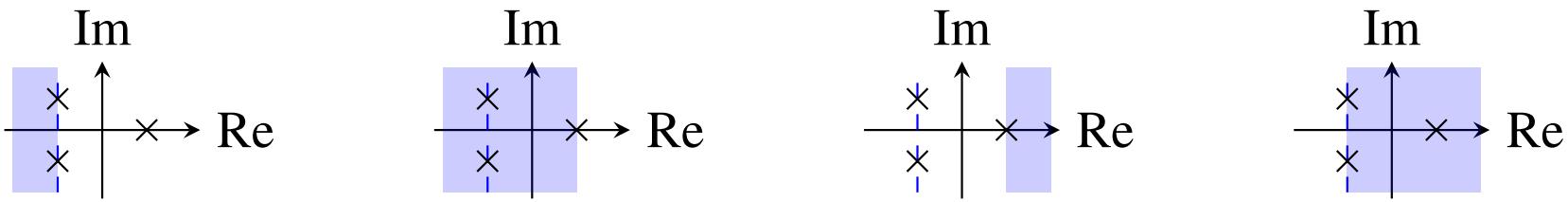
Invalid



Invalid

ROC Property 7: More on Rational Laplace Transforms

- If the Laplace transform X of a function x is ***rational*** (with at least one pole), then:
  1. If x is ***right sided***, the ROC of X is to the right of the rightmost pole of X (i.e., the ***RHP*** to the ***right of the rightmost pole***).
  2. If x is ***left sided***, the ROC of X is to the left of the leftmost pole of X (i.e., the ***LHP*** to the ***left of the leftmost pole***).
- This property is implied by properties 1, 2, 4, and 5.
- Some examples of sets that would be either valid or invalid as ROCs for X, if X is rational and x is left/right sided, are given below.

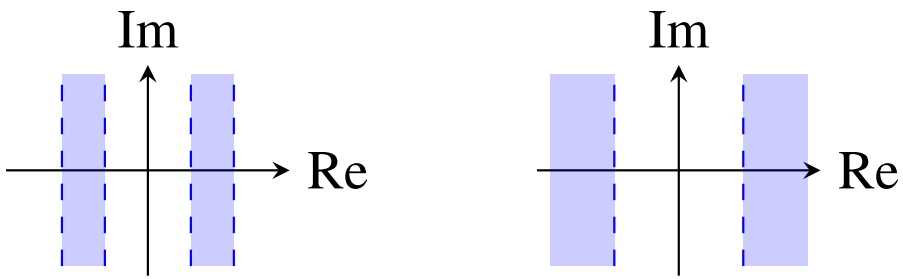


General Form of the ROC

- To summarize the results of properties 3, 4, 5, and 6, if the Laplace transform X of the function x exists, the ROC of X depends on the left- and right-sidedness of x as follows:

x		ROC of X
left sided	right sided	
no	no	strip
no	yes	RHP
yes	no	LHP
yes	yes	everywhere

- Thus, we can infer that, if X exists, its ROC can only be of the form of a LHP, a RHP, a vertical strip, or the entire complex plane.
- For example, the sets shown below would not be valid as ROCs.



Section 7.3

Properties of the Laplace Transform

Properties of the Laplace Transform

Property	Time Domain	Laplace Domain	ROC
Linearity	$a_1x_1(t)+a_2x_2(t)$	$a_1X_1(s)+a_2X_2(s)$	At least $R_1 \cap R_2$
Time-Domain Shifting	$x(t-t_0)$	$e^{-st_0}X(s)$	R
Laplace-Domain Shifting	$e^{s_0t}x(t)$	$X(s-s_0)$	$R+\mathrm{Re}(s_0)$
Time/Laplace-Domain Scaling	$x(at)$	$\frac{1}{\lvert a\rvert}X\left(\frac{s}{a}\right)$	aR
Conjugation	$x^*(t)$	$X^*(s^*)$	R
Time-Domain Convolution	$x_1 * x_2(t)$	$X_1(s)X_2(s)$	At least $R_1 \cap R_2$
Time-Domain Differentiation	$\frac{d}{dt}x(t)$	$sX(s)$	At least R
Laplace-Domain Differentiation	$-tx(t)$	$\frac{d}{ds}X(s)$	R
Time-Domain Integration	$\int_{-\infty}^{t} x(\tau)d\tau$	$\frac{1}{s}X(s)$	At least $R \cap \{\mathrm{Re}(s)>0\}$

Property	
Initial Value Theorem	$x(0^+)=\lim_{s\to\infty} sX(s)$
Final Value Theorem	$\lim_{t\to\infty} x(t)=\lim_{s\to 0} sX(s)$

Laplace Transform Pairs

Pair	$x(t)$	$X(s)$	ROC
1	$\delta(t)$	1	All s
2	$u(t)$	$\frac{1}{s}$	$\mathrm{Re}(s) > 0$
3	$-u(-t)$	$\frac{1}{s}$	$\mathrm{Re}(s) < 0$
4	$t^n u(t)$	$\frac{n!}{s^{n+1}}$	$\mathrm{Re}(s) > 0$
5	$-t^n u(-t)$	$\frac{n!}{s^{n+1}}$	$\mathrm{Re}(s) < 0$
6	$e^{-at} u(t)$	$\frac{1}{s+a}$	$\mathrm{Re}(s) > -a$
7	$-e^{-at} u(-t)$	$\frac{1}{s+a}$	$\mathrm{Re}(s) < -a$
8	$t^n e^{-at} u(t)$	$\frac{n!}{(s+a)^{n+1}}$	$\mathrm{Re}(s) > -a$
9	$-t^n e^{-at} u(-t)$	$\frac{n!}{(s+a)^{n+1}}$	$\mathrm{Re}(s) < -a$
10	$\cos(\omega_0 t) u(t)$	$\frac{s}{s^2+\omega_0^2}$	$\mathrm{Re}(s) > 0$
11	$\sin(\omega_0 t) u(t)$	$\frac{\omega_0}{s^2+\omega_0^2}$	$\mathrm{Re}(s) > 0$
12	$e^{-at} \cos(\omega_0 t) u(t)$	$\frac{s+a}{(s+a)^2+\omega_0^2}$	$\mathrm{Re}(s) > -a$
13	$e^{-at} \sin(\omega_0 t) u(t)$	$\frac{\omega_0}{(s+a)^2+\omega_0^2}$	$\mathrm{Re}(s) > -a$

Linearity

- If $x_1(t) \overset{\text{LT}}{\longleftrightarrow} X_1(s)$ with ROC R_1 and $x_2(t) \overset{\text{LT}}{\longleftrightarrow} X_2(s)$ with ROC R_2, then

  $$a_1 x_1(t) + a_2 x_2(t) \overset{\text{LT}}{\longleftrightarrow} a_1 X_1(s) + a_2 X_2(s)$$ with ROC R containing $R_1 \cap R_2$,

  where a_1 and a_2 are arbitrary complex constants.
- This is known as the **linearity property** of the Laplace transform.
- The ROC R always contains $R_1 \cap R_2$ but can be larger (in the case that pole-zero cancellation occurs).

Time-Domain Shifting

- If $x(t) \overset{\text{LT}}{\longleftrightarrow} X(s)$ with ROC R, then

$$x(t - t_0) \overset{\text{LT}}{\longleftrightarrow} e^{-st_0} X(s) \text{ with ROC } R,$$

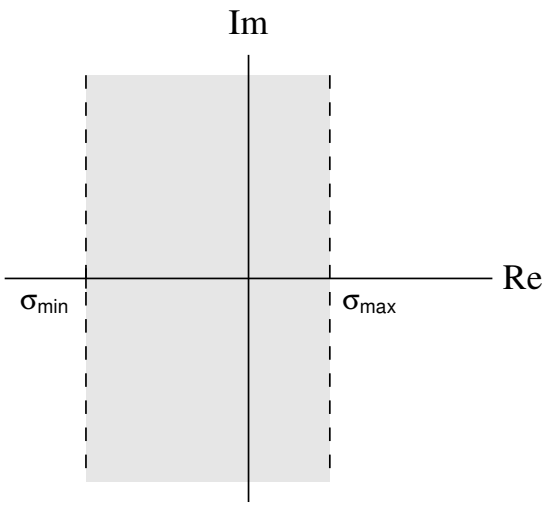
  where t_0 is an arbitrary real constant.
- This is known as the **time-domain shifting property** of the Laplace transform.

Laplace-Domain Shifting

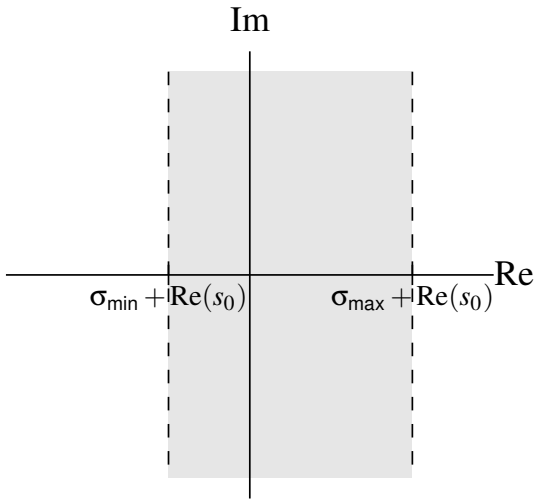
- If $x(t) \overset{\text{LT}}{\longleftrightarrow} X(s)$ with ROC R, then

$$e^{s_0 t} x(t) \overset{\text{LT}}{\longleftrightarrow} X(s - s_0) \text{ with ROC } R' = R + \text{Re}(s_0),$$

  where s_0 is an arbitrary complex constant.
- This is known as the **Laplace-domain shifting property** of the Laplace transform.
- As illustrated below, the ROC R is ***shifted*** right by $\text{Re}(s_0)$.



R



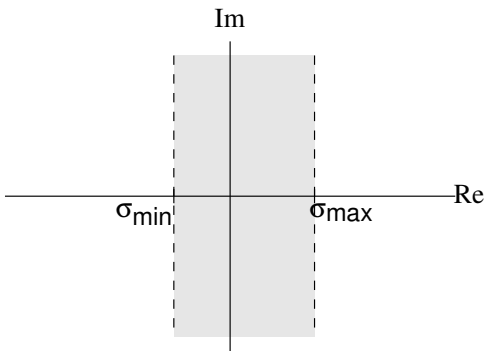
$R' = R + \text{Re}(s_0)$

Time-Domain/Laplace-Domain Scaling

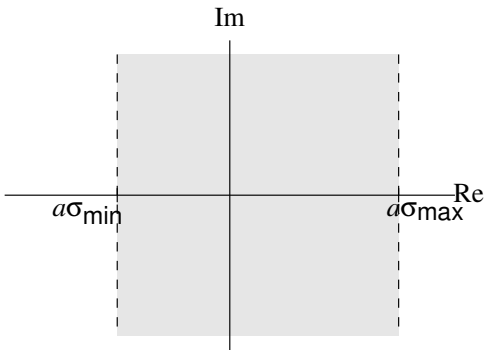
- If $x(t) \overset{\text{LT}}{\longleftrightarrow} X(s)$ with ROC R, then

$$x(at) \overset{\text{LT}}{\longleftrightarrow} \frac{1}{|a|} X\left(\frac{s}{a}\right) \quad \text{with ROC } R' = aR,$$

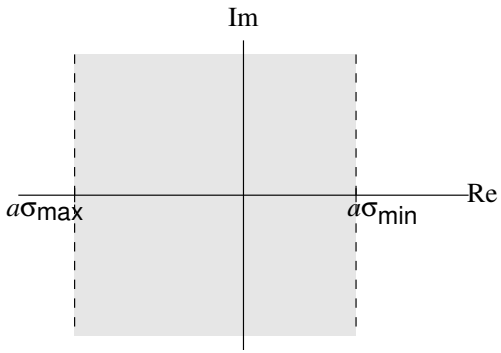
  where a is a nonzero real constant.
- This is known as the **(time-domain/Laplace-domain) scaling property** of the Laplace transform.
- As illustrated below, the ROC R is ***scaled*** and ***possibly flipped*** left to right.



R



$R' = aR,\ a > 0$



$R' = aR,\ a < 0$

Conjugation

- If $x(t) \overset{\text{LT}}{\longleftrightarrow} X(s)$ with ROC R, then

$$x^*(t) \overset{\text{LT}}{\longleftrightarrow} X^*(s^*) \text{ with ROC } R.$$

- This is known as the **conjugation property** of the Laplace transform.

Time-Domain Convolution

- If $x_1(t) \overset{\text{LT}}{\longleftrightarrow} X_1(s)$ with ROC R_1 and $x_2(t) \overset{\text{LT}}{\longleftrightarrow} X_2(s)$ with ROC R_2, then

$$x_1 * x_2(t) \overset{\text{LT}}{\longleftrightarrow} X_1(s)X_2(s) \text{ with ROC } R \text{ containing } R_1 \cap R_2.$$

- This is known as the **time-domain convolution property** of the Laplace transform.
- The ROC R always contains $R_1 \cap R_2$ but can be larger than this intersection (if pole-zero cancellation occurs).
- Convolution in the time domain becomes ***multiplication*** in the Laplace domain.
- Consequently, it is often much easier to work with LTI systems in the Laplace domain, rather than the time domain.

Time-Domain Differentiation

- If $x(t) \overset{\text{LT}}{\longleftrightarrow} X(s)$ with ROC R, then

$$\frac{dx(t)}{dt} \overset{\text{LT}}{\longleftrightarrow} sX(s) \text{ with ROC } R' \text{ containing } R.$$

- This is known as the **time-domain differentiation property** of the Laplace transform.
- The ROC R' always contains R but can be larger than R (if pole-zero cancellation occurs).
- Differentiation in the time domain becomes ***multiplication by*** s in the Laplace domain.
- Consequently, it can often be much easier to work with differential equations in the Laplace domain, rather than the time domain.

Laplace-Domain Differentiation

- If $x(t) \overset{\text{LT}}{\longleftrightarrow} X(s)$ with ROC R, then

$$-tx(t) \overset{\text{LT}}{\longleftrightarrow} \frac{dX(s)}{ds} \text{ with ROC } R.$$

- This is known as the **Laplace-domain differentiation property** of the Laplace transform.

Time-Domain Integration

- If $x(t) \overset{\text{LT}}{\longleftrightarrow} X(s)$ with ROC R, then

$$\int_{-\infty}^{t} x(\tau)d\tau \overset{\text{LT}}{\longleftrightarrow} \frac{1}{s}X(s) \text{ with ROC } R' \text{ containing } R \cap \{\text{Re}(s) > 0\}.$$

- This is known as the **time-domain integration property** of the Laplace transform.
- The ROC R' always contains at least $R \cap \{\text{Re}(s) > 0\}$ but can be larger (if pole-zero cancellation occurs).
- Integration in the time domain becomes ***division by*** s in the Laplace domain.
- Consequently, it is often much easier to work with integral equations in the Laplace domain, rather than the time domain.

Initial Value Theorem

- For a function x with Laplace transform X, if x is ***causal*** and contains ***no impulses or higher order singularities at the origin***, then

$$x(0^+) = \lim_{s \to \infty} sX(s),$$

  where $x(0^+)$ denotes the limit of $x(t)$ as t approaches zero from positive values of t.
- This result is known as the **initial value theorem**.
- In situations where X is known but x is not, the initial value theorem eliminates the need to explicitly find x by an inverse Laplace transform calculation in order to evaluate $x(0^+)$.
- In practice, the values of functions at the origin are frequently of interest, as such values often convey information about the initial state of systems.
- The initial value theorem can sometimes also be helpful in checking for errors in Laplace transform calculations.

Final Value Theorem

- For a function x with Laplace transform X, if x is ***causal*** and $x(t)$ has a ***finite limit*** as $t \to \infty$, then

$$\lim_{t \to \infty} x(t) = \lim_{s \to 0} sX(s).$$

- This result is known as the **final value theorem**.
- In situations where X is known but x is not, the final value theorem eliminates the need to explicitly find x by an inverse Laplace transform calculation in order to evaluate $\lim_{t \to \infty} x(t)$.
- In practice, the values of functions at infinity are frequently of interest, as such values often convey information about the steady-state behavior of systems.
- The final value theorem can sometimes also be helpful in checking for errors in Laplace transform calculations.

More Laplace Transform Examples

THIS SLIDE IS INTENTIONALLY LEFT BLANK.

Section 7.4

Determination of Inverse Laplace Transform

Finding Inverse Laplace Transform

- Recall that the inverse Laplace transform x of X is given by

$$x(t) = \frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} X(s) e^{st} ds,$$

  where $\mathrm{Re}(s) = \sigma$ is in the ROC of X.
- Unfortunately, the above contour integration can often be ***quite tedious*** to compute.
- Consequently, we do not usually compute the inverse Laplace transform directly using the above equation.
- For rational functions, the inverse Laplace transform can be more easily computed using ***partial fraction expansions***.
- Using a partial fraction expansion, we can express a rational function as a sum of lower-order rational functions whose inverse Laplace transforms can typically be found in tables.

Section 7.5

Laplace Transform and LTI Systems

System Function of LTI Systems

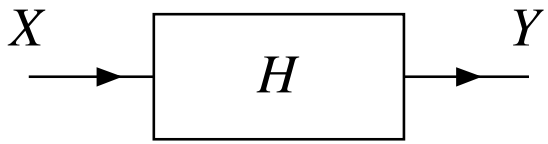
- Consider a LTI system with input x, output y, and impulse response h. Let X, Y, and H denote the Laplace transforms of x, y, and h, respectively.
- Since $y(t) = x * h(t)$, the system is characterized in the Laplace domain by

$$Y(s) = X(s)H(s).$$

- As a matter of terminology, we refer to H as the **system function** (or **transfer function**) of the system (i.e., the system function is the Laplace transform of the impulse response).
- A LTI system is ***completely characterized*** by its system function H.
- When viewed in the Laplace domain, a LTI system forms its output by multiplying its input with its system function.
- If the ROC of H includes the imaginary axis, then $H(j\omega)$ is the ***frequency response*** of the LTI system.

Block Diagram Representations of LTI Systems

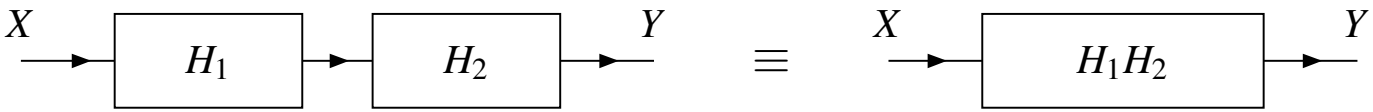
- Consider a LTI system with input x, output y, and impulse response h, and let X, Y, and H denote the Laplace transforms of x, y, and h, respectively.
- Often, it is convenient to represent such a system in block diagram form in the Laplace domain as shown below.



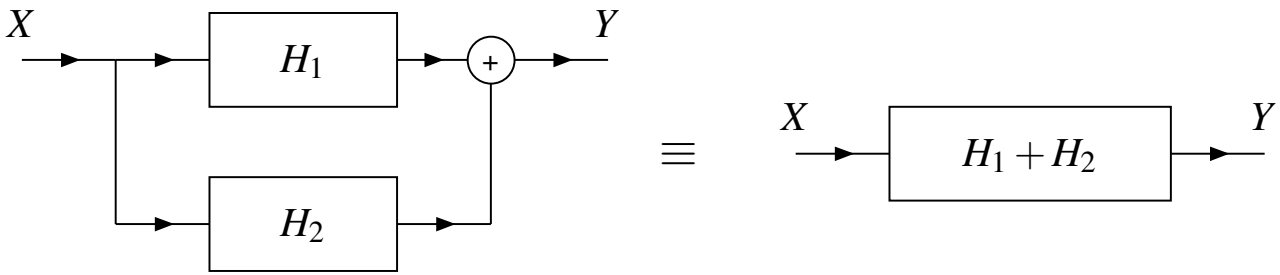
- Since a LTI system is completely characterized by its system function, we typically label the system with this quantity.

Interconnection of LTI Systems

- The ***series*** interconnection of the LTI systems with system functions H_1 and H_2 is the LTI system with system function $H_1 H_2$. That is, we have the equivalence shown below.



- The ***parallel*** interconnection of the LTI systems with system functions H_1 and H_2 is the LTI system with the system function $H_1 + H_2$. That is, we have the equivalence shown below.



Causality

- If a LTI system is ***causal***, its impulse response is causal, and therefore ***right sided***. From this, we have the result below.
- **Theorem.** The ROC associated with the system function of a ***causal*** LTI system is a ***RHP*** or the ***entire complex plane***.
- In general, the ***converse*** of the above theorem is ***not necessarily true***. That is, if the ROC of the system function is a RHP or the entire complex plane, it is not necessarily true that the system is causal.
- If the system function is ***rational***, however, we have that the converse does hold, as indicated by the theorem below.
- **Theorem.** For a LTI system with a ***rational*** system function H, ***causality*** of the system is ***equivalent*** to the ROC of H being the ***RHP to the right of the rightmost pole*** or, if H has no poles, the entire complex plane.

BIBO Stability

- Whether or not a system is BIBO stable depends on the ROC of its system function.
- **Theorem.** A LTI system is ***BIBO stable*** if and only if the ROC of its system function H contains the ***imaginary axis*** (i.e., $\mathrm{Re}(s) = 0$).
- **Theorem.** A ***causal*** LTI system with a (proper) ***rational*** system function H is BIBO stable if and only if all of the poles of H lie in the left half of the plane (i.e., all of the poles have ***negative real parts***).

Invertibility

- A LTI system $\mathcal{H}$ with system function H is invertible if and only if there exists another LTI system with system function H_{inv} such that

$$H(s)H_{\text{inv}}(s) = 1,$$

  in which case H_{inv} is the system function of $\mathcal{H}^{-1}$ and

$$H_{\text{inv}}(s) = \frac{1}{H(s)}.$$

- Since distinct systems can have identical system functions (but with differing ROCs), the inverse of a LTI system is ***not necessarily unique***.
- In practice, however, we often desire a stable and/or causal system. So, although multiple inverse systems may exist, we are frequently only interested in ***one specific choice*** of inverse system (due to these additional constraints of stability and/or causality).

LTI Systems and Differential Equations

- Many LTI systems of practical interest can be represented using an ***Nth-order linear differential equation with constant coefficients***.
- Consider a system with input x and output y that is characterized by an equation of the form

$$\sum_{k=0}^{N} b_k \left(\frac{d}{dt}\right)^k y(t) = \sum_{k=0}^{M} a_k \left(\frac{d}{dt}\right)^k x(t),$$

  where the a_k and b_k are complex constants and $M \leq N$.
- Let h denote the impulse response of the system, and let X, Y, and H denote the Laplace transforms of x, y, and h, respectively.
- One can show that H is given by

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\sum_{k=0}^{M} a_k s^k}{\sum_{k=0}^{N} b_k s^k}.$$

- Observe that, for a system of the form considered above, the system function is always ***rational***.

Section 7.6

Application: Circuit Analysis

Electronic Circuits

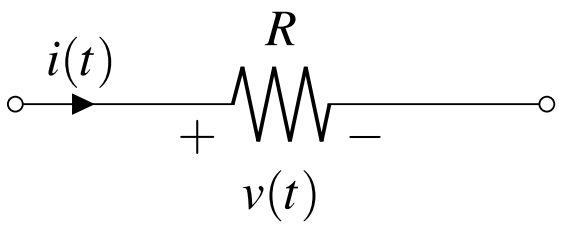
- An **electronic circuit** is a network of one or more interconnected circuit elements.
- The three most basic types of circuit elements are:
  1. resistors;
  2. inductors; and
  3. capacitors.
- Two fundamental quantities of interest in electronic circuits are current and voltage.
- **Current** is the rate at which electric charge flows through some part of a circuit, such as a circuit element, and is measured in units of amperes (A).
- **Voltage** is the difference in electric potential between two points in a circuit, such as across a circuit element, and is measured in units of volts (V).
- Voltage is essentially a force that makes electric charge (or current) flow.

Resistors

- A **resistor** is a circuit element that opposes the flow of current.
- A resistor is characterized by an equation of the form

$$v(t) = Ri(t) \quad \left(\text{or equivalently, } i(t) = \tfrac{1}{R}v(t)\right),$$

  where R is a nonnegative real constant, and v and i respectively denote the voltage across and current through the resistor as a function of time.
- As a matter of terminology, the quantity R is known as the **resistance** of the resistor.
- Resistance is measured in units of ohms (Ω).
- In circuit diagrams, a resistor is denoted by the symbol shown below.

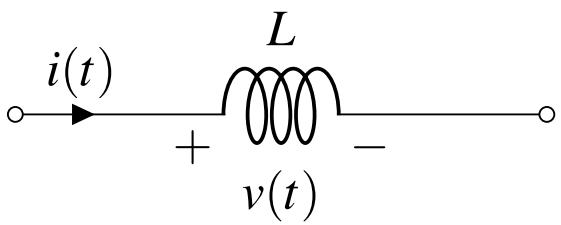


Inductors

- An **inductor** is a circuit element that converts an electric current into a magnetic field and vice versa.
- An inductor uses the energy stored in a magnetic field in order to ***oppose changes in current*** (through the inductor).
- An inductor is characterized by an equation of the form

$$v(t) = L\frac{d}{dt}i(t) \quad \text{(or equivalently, } i(t) = \frac{1}{L}\int_{-\infty}^{t} v(\tau)d\tau\text{)},$$

  where L is a nonnegative real constant, and v and i respectively denote the voltage across and current through the inductor as a function of time.
- As a matter of terminology, the quantity L is known as the **inductance** of the inductor.
- Inductance is measured in units of henrys (H).
- In circuit diagrams, an inductor is denoted by the symbol shown below.

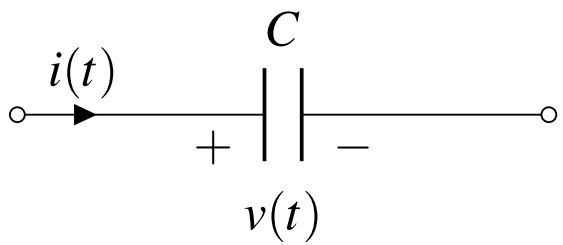


Capacitors

- A **capacitor** is a circuit element that stores electric charge.
- A capacitor uses the energy stored in an electric field in order to ***oppose changes in voltage*** (across the capacitor).
- A capacitor is characterized by an equation of the form

$$v(t) = \frac{1}{C}\int_{-\infty}^{t} i(\tau)d\tau \quad \text{(or equivalently, } i(t) = C\frac{d}{dt}v(t)\text{)},$$

  where C is a nonnegative real constant, and v and i respectively denote the voltage across and current through the capacitor as a function of time.
- As a matter of terminology, the quantity C is known as the **capacitance** of the capacitor.
- Capacitance is measured in units of farads (F).
- In circuit diagrams, a capacitor is denoted by the symbol shown below.



Circuit Analysis with the Laplace Transform

- The Laplace transform is a very useful tool for circuit analysis.
- The utility of the Laplace transform is partly due to the fact that the ***differential/integral*** equations that describe inductors and capacitors are much simpler to express in the Laplace domain than in the time domain.
- Let v and i denote the voltage across and current through a circuit element, and let V and I denote the Laplace transforms of v and i, respectively.
- In the Laplace domain, the equations characterizing a resistor, an inductor, and a capacitor respectively become:

$$V(s) = RI(s) \quad \text{(or equivalently, } I(s) = \tfrac{1}{R}V(s)\text{)};$$
$$V(s) = sLI(s) \quad \text{(or equivalently, } I(s) = \tfrac{1}{sL}V(s)\text{)}; \quad \text{and}$$
$$V(s) = \tfrac{1}{sC}I(s) \quad \text{(or equivalently, } I(s) = sCV(s)\text{)}.$$

- Note the absence of differentiation and integration in the above equations for an inductor and a capacitor.

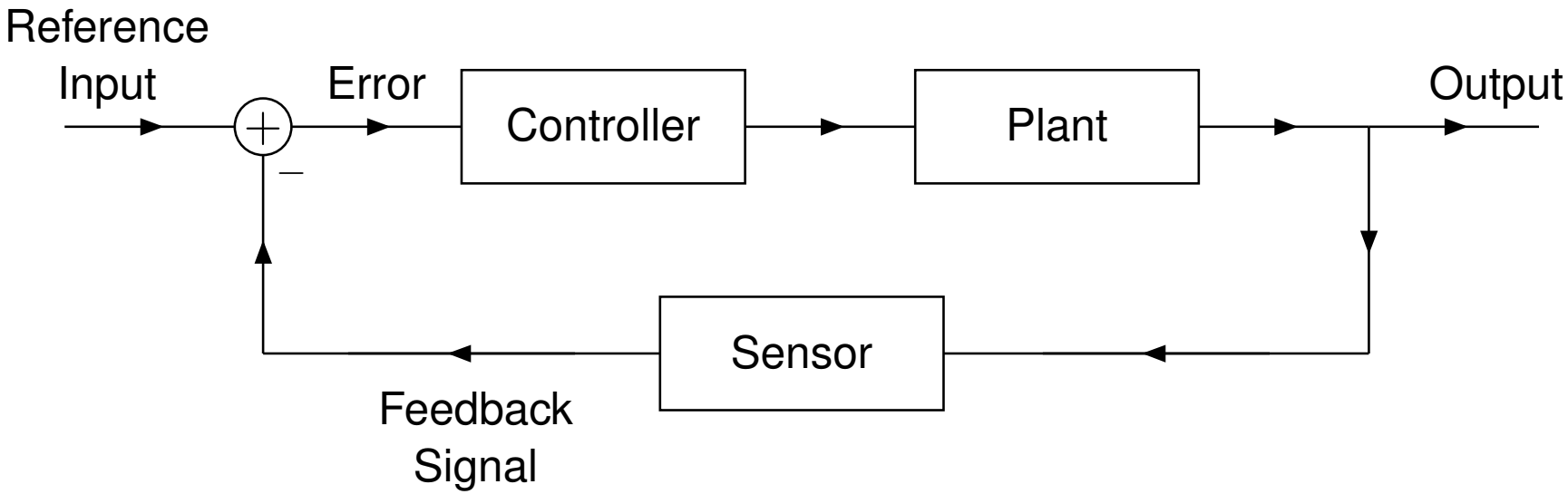
Section 7.7

Application: Design and Analysis of Control Systems

Control Systems

- A control system manages the behavior of one or more other systems with some specific goal.
- Typically, the goal is to force one or more physical quantities to assume particular desired values, where such quantities might include: positions, velocities, accelerations, forces, torques, temperatures, or pressures.
- The ***desired*** values of the quantities being controlled are collectively viewed as the ***input*** of the control system.
- The ***actual*** values of the quantities being controlled are collectively viewed as the ***output*** of the control system.
- A control system whose behavior is not influenced by the actual values of the quantities being controlled is called an **open loop** (or **non-feedback**) system.
- A control system whose behavior is influenced by the actual values of the quantities being controlled is called a **closed loop** (or **feedback**) system.
- An example of a simple control system would be a thermostat system, which controls the temperature in a room or building.

Feedback Control Systems



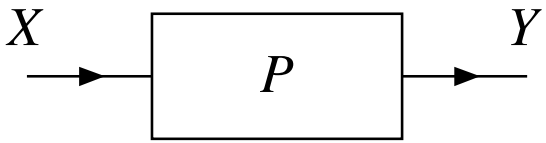
- **input**: ***desired value*** of the quantity to be controlled
- **output**: ***actual value*** of the quantity to be controlled
- **error**: ***difference*** between the desired and actual values
- **plant**: system to be controlled
- **sensor**: device used to measure the actual output
- **controller**: device that monitors the error and changes the input of the plant with the goal of forcing the error to zero

Stability Analysis of Feedback Systems

- Often, we want to ensure that a system is BIBO stable.
- The BIBO stability property is more easily characterized in the Laplace domain than in the time domain.
- Therefore, the Laplace domain is extremely useful for the stability analysis of systems.

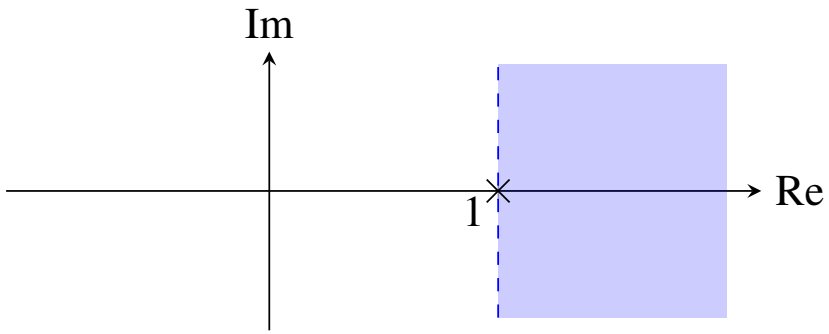
Stabilization Example: Unstable Plant

- causal LTI plant:



$$P(s) = \frac{10}{s-1}$$

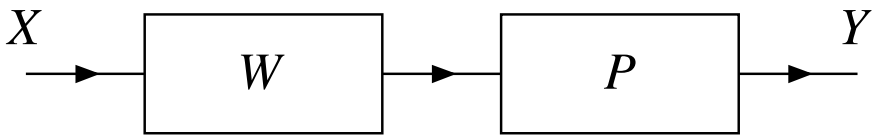
- ROC of P:



- system is not BIBO stable

Stabilization Example: Using Pole-Zero Cancellation

- system formed by series interconnection of plant and causal LTI compensator:

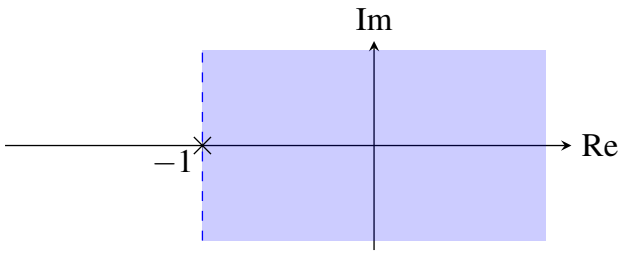


$$P(s) = \frac{10}{s-1}, \quad W(s) = \frac{s-1}{10(s+1)}$$

- system function H of overall system:

$$H(s) = W(s)P(s) = \left(\frac{s-1}{10(s+1)}\right)\left(\frac{10}{s-1}\right) = \frac{1}{s+1}$$

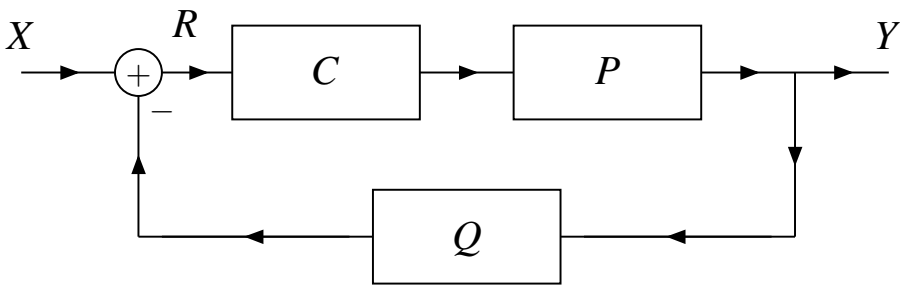
- ROC of H:



- overall system is BIBO stable

Stabilization Example: Using Feedback (1)

- feedback system (with causal LTI compensator and sensor):

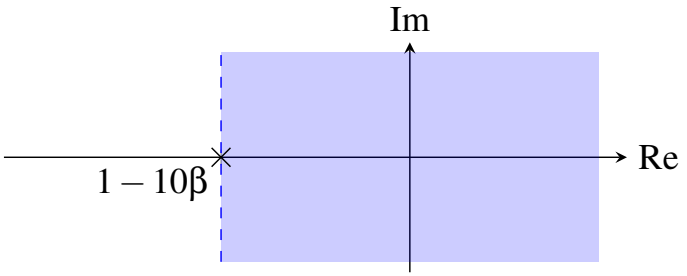


$$P(s) = \frac{10}{s-1}, \quad C(s) = \beta, \quad Q(s) = 1$$

- system function H of feedback system:

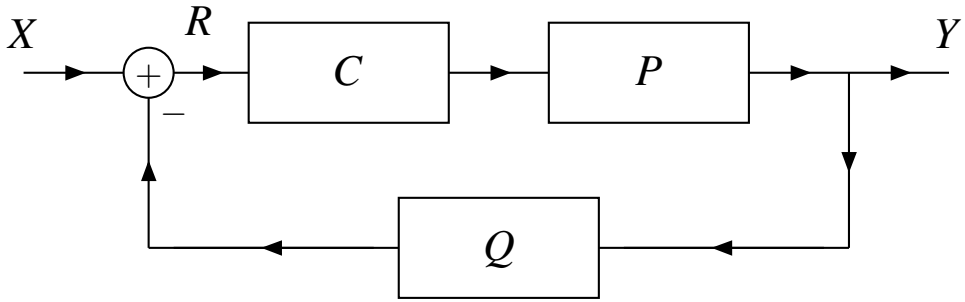
$$H(s) = \frac{C(s)P(s)}{1+C(s)P(s)Q(s)} = \frac{10\beta}{s-(1-10\beta)}$$

- ROC of H:



- feedback system is BIBO stable if and only if $1 - 10\beta < 0$ or equivalently $\beta > \frac{1}{10}$

Stabilization Example: Using Feedback (2)



$$R(s) = X(s) - Q(s)Y(s)$$
$$Y(s) = C(s)P(s)R(s)$$

$$\begin{aligned} Y(s) &= C(s)P(s)R(s) \\ &= C(s)P(s)[X(s) - Q(s)Y(s)] \\ &= C(s)P(s)X(s) - C(s)P(s)Q(s)Y(s) \end{aligned}$$

$$[1 + C(s)P(s)Q(s)]Y(s) = C(s)P(s)X(s)$$
$$H(s) = \frac{Y(s)}{X(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)Q(s)}$$

Stabilization Example: Using Feedback (3)

$$P(s) = \frac{10}{s-1}, \quad C(s) = \beta, \quad Q(s) = 1$$

$$\begin{aligned} H(s) &= \frac{C(s)P(s)}{1 + C(s)P(s)Q(s)} \\ &= \frac{\beta(\frac{10}{s-1})}{1 + \beta(\frac{10}{s-1})(1)} \\ &= \frac{10\beta}{s - 1 + 10\beta} \\ &= \frac{10\beta}{s - (1 - 10\beta)} \end{aligned}$$

Remarks on Stabilization Via Pole-Zero Cancellation

- Pole-zero cancellation is not achievable in practice, and therefore it cannot be used to stabilize real-world systems.
- The theoretical models used to represent real-world systems are only approximations due to many factors, including the following:
  - Determining the system function of a system involves measurement, which always has some error.
  - A system cannot be built with such precision that it will have exactly some prescribed system function.
  - The system function of most systems will vary at least slightly with changes in the physical environment.
  - Although a LTI model is used to represent a system, the likely reality is that the system is not exactly LTI, which introduces error.
- Due to approximation error, the effective poles and zeros of the system function will only be approximately where they are expected to be.
- Since pole-zero cancellation requires that a pole and zero be placed at exactly the same location, any error will prevent this cancellation from being achieved.

Section 7.8

Unilateral Laplace Transform

Unilateral Laplace Transform

- The **unilateral Laplace transform** of the function x, denoted $\mathcal{L}_{\text{u}}x$ or X, is defined as

$$\mathcal{L}_{\text{u}}x(s) = X(s) = \int_{0^-}^{\infty} x(t)e^{-st}dt.$$

- The unilateral Laplace transform is related to the bilateral Laplace transform as follows:

$$\mathcal{L}_{\text{u}}x(s) = \int_{0^-}^{\infty} x(t)e^{-st}dt = \int_{-\infty}^{\infty} x(t)u(t)e^{-st}dt = \mathcal{L}\{xu\}(s).$$

- In other words, the unilateral Laplace transform of the function x is simply the bilateral Laplace transform of the function xu.
- Since $\mathcal{L}_{\text{u}}x = \mathcal{L}\{xu\}$ and xu is always a ***right-sided*** function, the ROC associated with $\mathcal{L}_{\text{u}}x$ is always either a ***RHP*** or the ***entire complex plane***.
- For this reason, we often ***do not explicitly indicate the ROC*** when working with the unilateral Laplace transform.

Inversion of the Unilateral Laplace Transform

- With the unilateral Laplace transform, the same inverse transform equation is used as in the bilateral case.
- The unilateral Laplace transform is ***only invertible for causal functions***.
- In particular, we have

$$\begin{aligned}\mathcal{L}_{\mathrm{u}}^{-1}\{\mathcal{L}_{\mathrm{u}}x\}(t) &= \mathcal{L}_{\mathrm{u}}^{-1}\{\mathcal{L}\{xu\}\}(t)\\ &= \mathcal{L}^{-1}\{\mathcal{L}\{xu\}\}(t)\\ &= x(t)u(t)\\ &= \begin{cases} x(t) & t \geq 0\\ 0 & t < 0. \end{cases}\end{aligned}$$

- For a noncausal function x, we can only recover $x(t)$ for $t \geq 0$.

Unilateral Versus Bilateral Laplace Transform

- Due to the close relationship between the unilateral and bilateral Laplace transforms, these two transforms have some similarities in their properties.
- Since these two transforms are not identical, however, their properties differ in some cases, often in subtle ways.
- In the unilateral case, we have that:
  1. the time-domain convolution property has the additional requirement that the functions being convolved must be ***causal***;
  2. the time/Laplace-domain scaling property has the additional constraint that the scaling factor must be ***positive***;
  3. the time-domain differentiation property has an ***extra term*** in the expression for $\mathcal{L}_{\mathrm{u}}\{\mathcal{D}x\}(t)$, where $\mathcal{D}$ denotes the derivative operator (namely, $-x(0^-)$);
  4. the time-domain integration property has a ***different lower limit*** in the time-domain integral (namely, 0^- instead of $-\infty$); and
  5. the time-domain shifting property ***does not hold*** (except in special circumstances).

Properties of the Unilateral Laplace Transform

Property	Time Domain	Laplace Domain
Linearity	$a_1 x_1(t) + a_2 x_2(t)$	$a_1 X_1(s) + a_2 X_2(s)$
Laplace-Domain Shifting	$e^{s_0 t} x(t)$	$X(s - s_0)$
Time/Laplace-Domain Scaling	$x(at)$, $a > 0$	$\frac{1}{a} X\left(\frac{s}{a}\right)$
Conjugation	$x^*(t)$	$X^*(s^*)$
Time-Domain Convolution	$x_1 * x_2(t)$, x_1 and x_2 are causal	$X_1(s) X_2(s)$
Time-Domain Differentiation	$\frac{d}{dt} x(t)$	$sX(s) - x(0^-)$
Laplace-Domain Differentiation	$-tx(t)$	$\frac{d}{ds} X(s)$
Time-Domain Integration	$\int_{0^-}^{t} x(\tau) d\tau$	$\frac{1}{s} X(s)$

Property	
Initial Value Theorem	$x(0^+) = \lim_{s \to \infty} sX(s)$
Final Value Theorem	$\lim_{t \to \infty} x(t) = \lim_{s \to 0} sX(s)$

Unilateral Laplace Transform Pairs

Pair	$x(t),\ t \geq 0$	$X(s)$
1	$\delta(t)$	1
2	1	$\frac{1}{s}$
3	t^n	$\frac{n!}{s^{n+1}}$
4	e^{-at}	$\frac{1}{s+a}$
5	$t^n e^{-at}$	$\frac{n!}{(s+a)^{n+1}}$
6	$\cos(\omega_0 t)$	$\frac{s}{s^2+\omega_0^2}$
7	$\sin(\omega_0 t)$	$\frac{\omega_0}{s^2+\omega_0^2}$
8	$e^{-at}\cos(\omega_0 t)$	$\frac{s+a}{(s+a)^2+\omega_0^2}$
9	$e^{-at}\sin(\omega_0 t)$	$\frac{\omega_0}{(s+a)^2+\omega_0^2}$

Solving Differential Equations [Using the Unilateral Laplace Transform]

- Many systems of interest in engineering applications can be characterized by constant-coefficient linear differential equations.
- One common use of the unilateral Laplace transform is in solving constant-coefficient linear differential equations with nonzero initial conditions.

Part 8

Discrete-Time (DT) Signals and Systems

Section 8.1

Independent- and Dependent-Variable Transformations

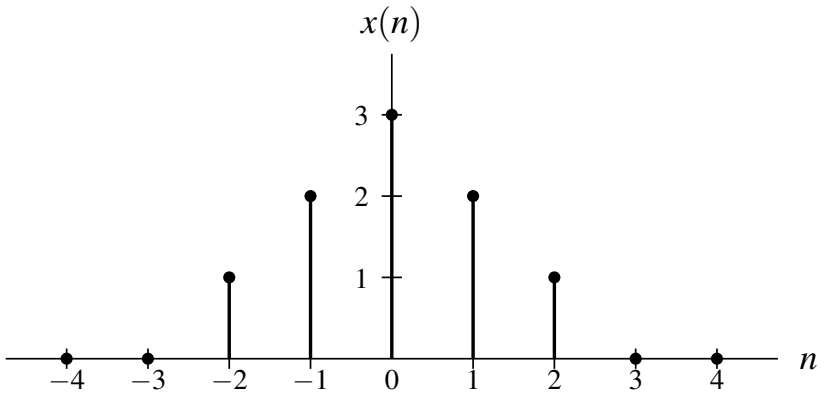
Time Shifting (Translation)

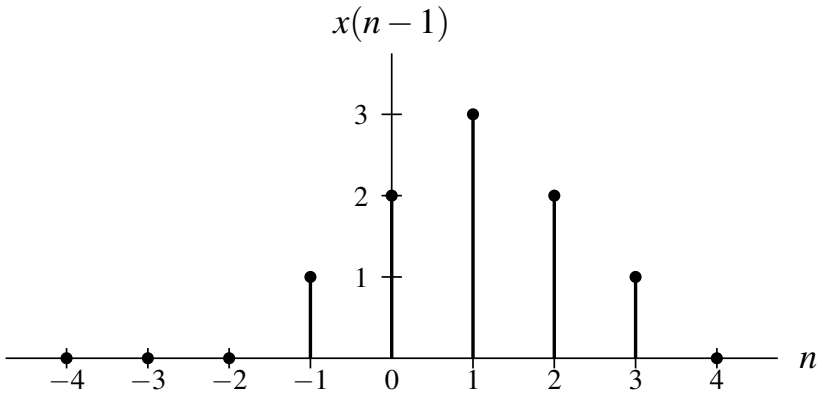
- **Time shifting** (also called **translation**) maps the input sequence x to the output sequence y as given by

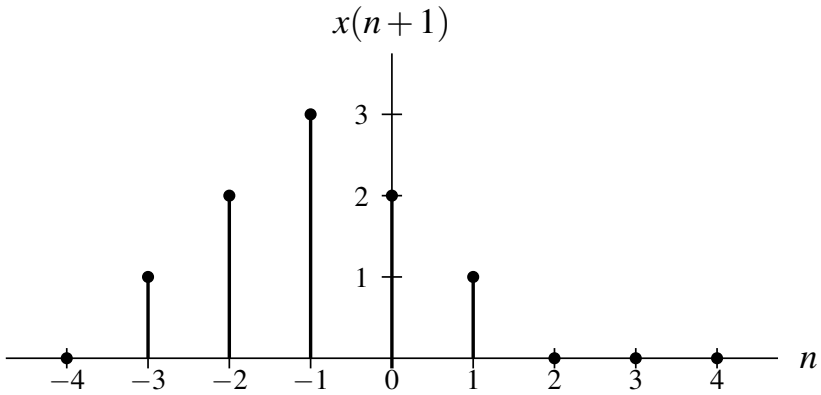
$$y(n) = x(n-b),$$

  where b is an integer.
- Such a transformation shifts the sequence (to the left or right) along the time axis.
- If $b > 0$, y is ***shifted to the right*** by $|b|$, relative to x (i.e., delayed in time).
- If $b < 0$, y is ***shifted to the left*** by $|b|$, relative to x (i.e., advanced in time).

Time Shifting (Translation): Example





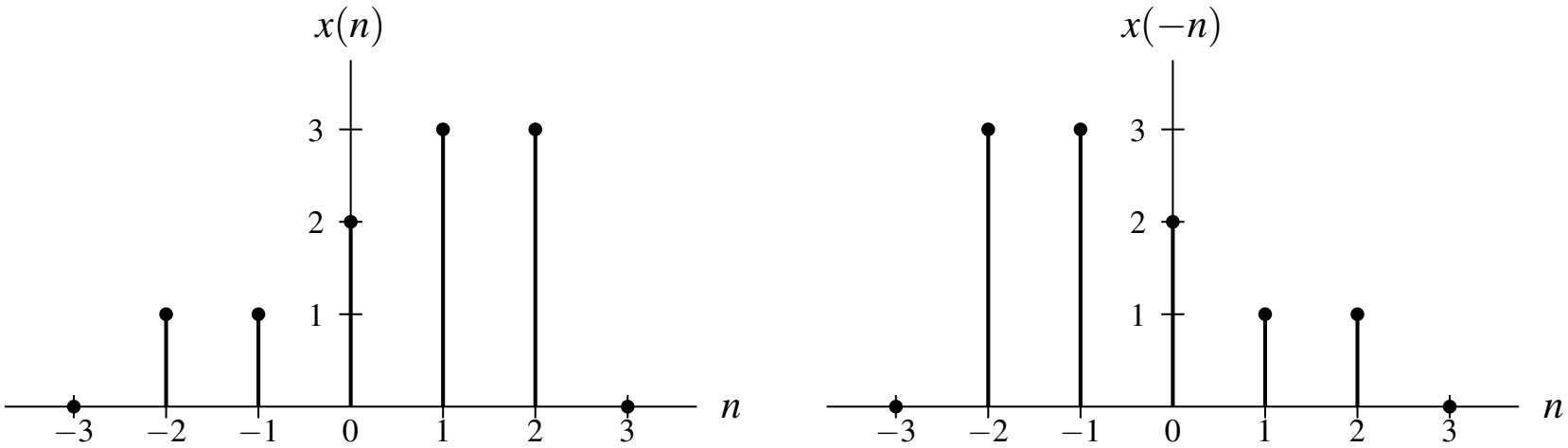


Time Reversal (Reflection)

- **Time reversal** (also known as **reflection**) maps the input sequence x to the output sequence y as given by

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

- Geometrically, the output sequence y is a reflection of the input sequence x about the (vertical) line $n = 0$.



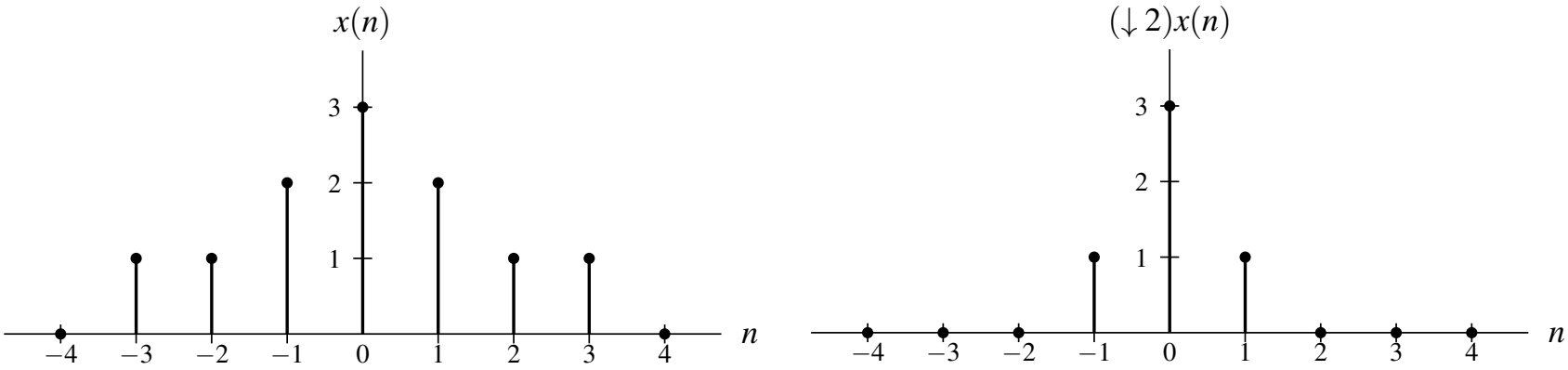
Downsampling

- **Downsampling** maps the input sequence x to the output sequence y as given by

$$y(n) = (\downarrow a)x(n) = x(an),$$

where a is a ***strictly positive*** integer.

- The output sequence y is produced from the input sequence x by keeping only every ath sample of x.



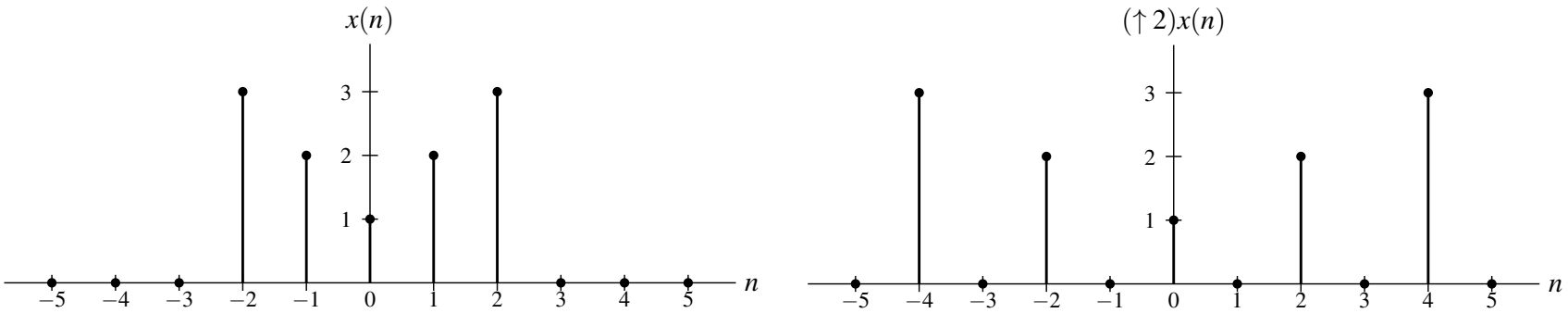
Upsampling

- **Upsampling** maps the input sequence x to the output sequence y as given by

$$y(n) = (\uparrow a)x(n) = \begin{cases} x(n/a) & n/a \text{ is an integer} \\ 0 & \text{otherwise,} \end{cases}$$

  where a is a strictly positive integer.

- The output sequence y is produced from the input sequence x by inserting $a-1$ zeros between all of the samples of x.



Combined Independent-Variable Transformations

- Consider a transformation that maps the input sequence x to the output sequence y as given by

$$y(n) = x(an - b),$$

  where a and b are integers and $a \neq 0$.
- Such a transformation is a combination of time shifting, downsampling, and time reversal operations.
- Time reversal ***commutes*** with downsampling.
- Time shifting ***does not commute*** with time reversal or downsampling.
- The above transformation is equivalent to:
  1. first, time shifting x by b;
  2. then, downsampling the result by $|a|$ and, if $a < 0$, time reversing as well.
- If $\frac{b}{a}$ is an integer, the above transformation is also equivalent to:
  1. first, downsampling x by $|a|$ and, if $a < 0$, time reversing;
  2. then, time shifting the result by $\frac{b}{a}$.
- Note that the time shift is not by the same amount in both cases.

Section 8.2

Properties of Sequences

Symmetry and Addition/Multiplication

- Sums involving even and odd sequences have the following properties:
  - The sum of two even sequences is even.
  - The sum of two odd sequences is odd.
  - The sum of an even sequence and odd sequence is neither even nor odd, provided that neither of the sequences is identically zero.
- That is, the ***sum*** of sequences with the ***same type of symmetry*** also has the ***same type of symmetry***.
- Products involving even and odd sequences have the following properties:
  - The product of two even sequences is even.
  - The product of two odd sequences is even.
  - The product of an even sequence and an odd sequence is odd.
- That is, the ***product*** of sequences with the ***same type of symmetry*** is ***even***, while the ***product*** of sequences with ***opposite types of symmetry*** is ***odd***.

Decomposition of a Sequence into Even and Odd Parts

- Every sequence x has a ***unique*** representation of the form

$$x(n) = x_{\mathrm{e}}(n) + x_{\mathrm{o}}(n),$$

  where the sequences x_{e} and x_{o} are ***even*** and ***odd***, respectively.
- In particular, the sequences x_{e} and x_{o} are given by

$$x_{\mathrm{e}}(n) = \frac{1}{2}\left[x(n) + x(-n)\right] \quad \text{and} \quad x_{\mathrm{o}}(n) = \frac{1}{2}\left[x(n) - x(-n)\right].$$

- The sequences x_{e} and x_{o} are called the **even part** and **odd part** of x, respectively.
- For convenience, the even and odd parts of x are often denoted as $\mathrm{Even}\{x\}$ and $\mathrm{Odd}\{x\}$, respectively.

Sum of Periodic Sequences

- The **least common multiple** of two (strictly positive) integers a and b, denoted $\text{lcm}(a,b)$, is the smallest positive integer that is divisible by both a and b.
- The quantity $\text{lcm}(a,b)$ can be easily determined from a prime factorization of the integers a and b by taking the product of the highest power for each prime factor appearing in these factorizations. Example:

$$\text{lcm}(20,6) = \text{lcm}(2^2 \cdot 5^1, 2^1 \cdot 3^1) = 2^2 \cdot 3^1 \cdot 5^1 = 60;$$
$$\text{lcm}(54,24) = \text{lcm}(2^1 \cdot 3^3, 2^3 \cdot 3^1) = 2^3 \cdot 3^3 = 216; \quad \text{and}$$
$$\text{lcm}(24,90) = \text{lcm}(2^3 \cdot 3^1, 2^1 \cdot 3^2 \cdot 5^1) = 2^3 \cdot 3^2 \cdot 5^1 = 360.$$

- **Sum of periodic sequences.** For any two periodic sequences x_1 and x_2 with fundamental periods N_1 and N_2, respectively, the sum $x_1 + x_2$ is ***periodic*** with period $\text{lcm}(N_1, N_2)$.

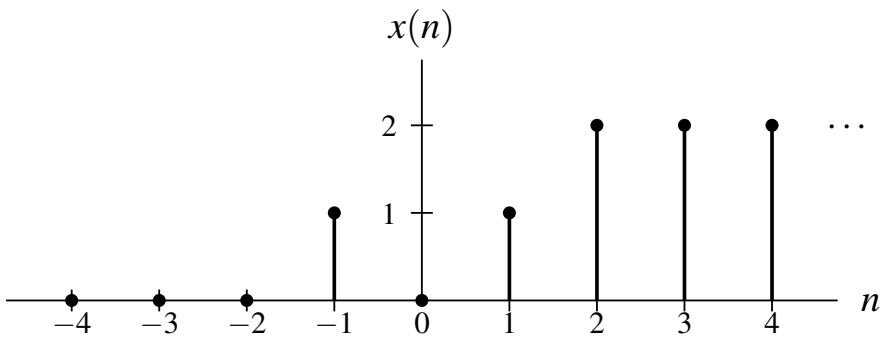
Right-Sided Sequences

- A sequence x is said to be **right sided** if, for some (finite) integer constant n_0, the following condition holds:

$$x(n) = 0 \quad \text{for all } n < n_0$$

(i.e., x is ***only potentially nonzero to the right of*** n_0).

- An example of a right-sided sequence is shown below.



- A sequence x is said to be **causal** if

$$x(n) = 0 \quad \text{for all } n < 0.$$

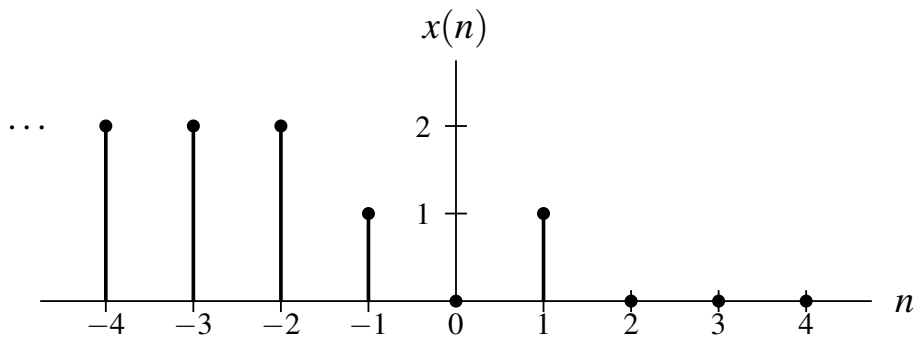
- A causal sequence is a ***special case*** of a right-sided sequence.
- A causal sequence is not to be confused with a causal system. In these two contexts, the word "causal" has very different meanings.

Left-Sided Sequences

- A sequence x is said to be **left sided** if, for some (finite) integer constant n_0, the following condition holds:

$$x(n) = 0 \quad \text{for all } n > n_0$$

  (i.e., x is ***only potentially nonzero to the left of*** n_0).
- An example of a left-sided sequence is shown below.



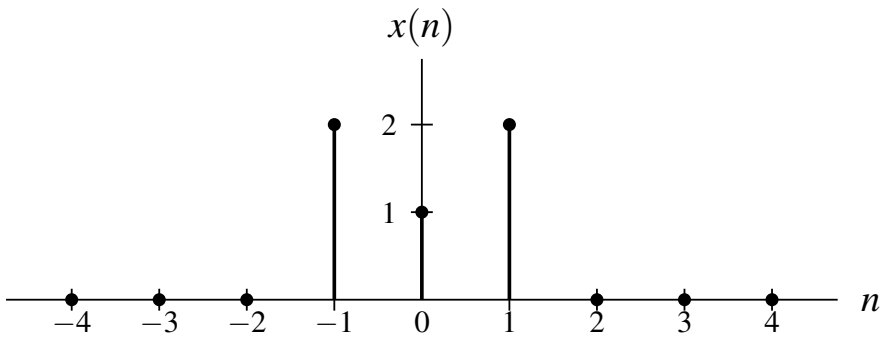
- A sequence x is said to be **anticausal** if

$$x(n) = 0 \quad \text{for all } n > 0.$$

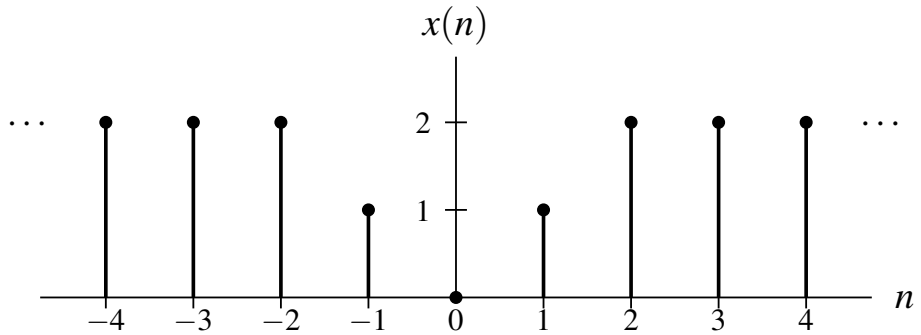
- An anticausal sequence is a ***special case*** of a left-sided sequence.
- An anticausal sequence is not to be confused with an anticausal system. In these two contexts, the word "anticausal" has very different meanings.

Finite-Duration and Two-Sided Sequences

- A sequence that is both left sided and right sided is said to be **finite duration** (or **time limited**).
- An example of a finite-duration sequence is shown below.



- A sequence that is neither left sided nor right sided is said to be **two sided**.
- An example of a two-sided sequence is shown below.



Bounded Sequences

- A sequence x is said to be **bounded** if there exists some (***finite***) positive real constant A such that

$$|x(n)| \leq A \quad \text{for all } n$$

  (i.e., $x(n)$ is ***finite*** for all n).
- Examples of bounded sequences include any constant sequence.
- Examples of unbounded sequences include any nonconstant polynomial sequence.

Energy of a Sequence

- The **energy** E contained in the sequence x is given by

$$E = \sum_{k=-\infty}^{\infty} |x(k)|^2 .$$

- A signal with finite energy is said to be an **energy signal**.

Section 8.3

Elementary Sequences

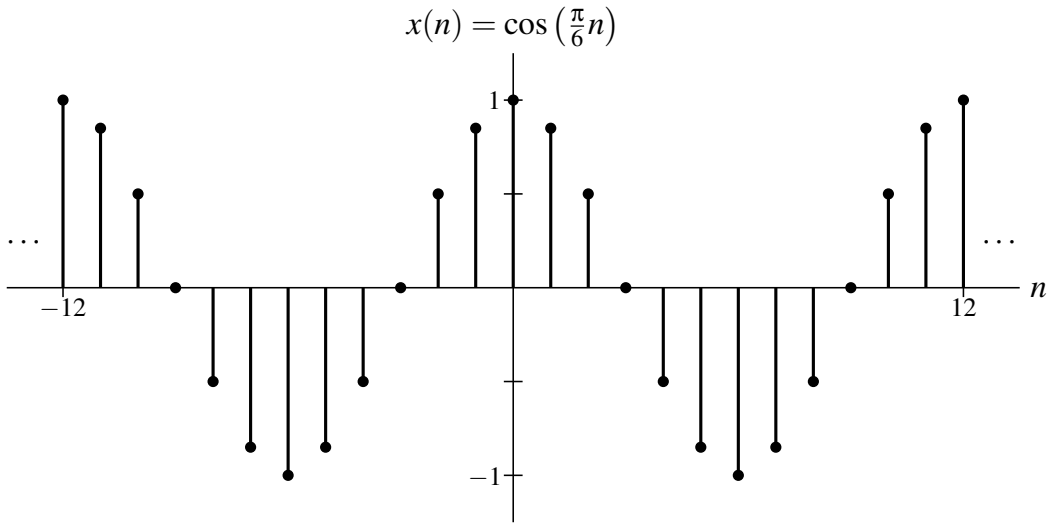
Real Sinusoidal Sequences

- A **real sinusoidal sequence** is a sequence of the form

$$x(n) = A\cos(\Omega n + \theta),$$

where A, Ω, and θ are ***real*** constants.

- A real sinusoid is ***periodic*** if and only if $\frac{\Omega}{2\pi}$ is a ***rational number***, in which case the fundamental period is the ***smallest integer*** of the form $\frac{2\pi k}{|\Omega|}$ where k is a (strictly) positive integer.
- For all integer k, $x_k(n) = A\cos([\Omega + 2\pi k]n + \theta)$ is the ***same*** sequence.
- An example of a periodic real sinusoid with fundamental period 12 is shown plotted below.



Oscillation Rate of Real Sinusoidal Sequences

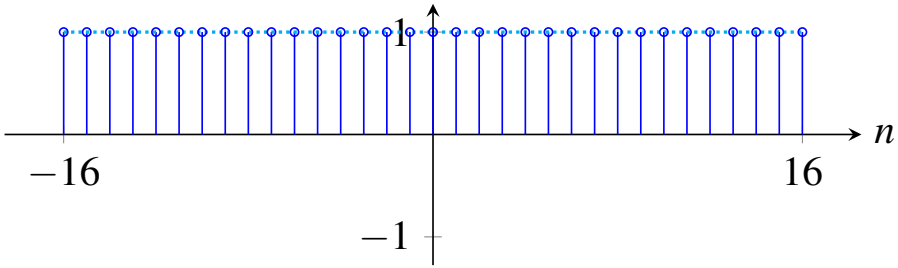
- Unlike their continuous-time counterparts, real sinusoidal sequences have an upper bound on the rate at which they can oscillate.
- Since $x_k(n) = A\cos([\Omega + 2\pi k]n + \theta)$ is the same sequence for all integer k, we consider only $0 \leq \Omega < 2\pi$ without loss of generality.
- Consider the set of real sinusoidal sequences of the form

$$x(n) = A\cos(\Omega n + \theta),$$

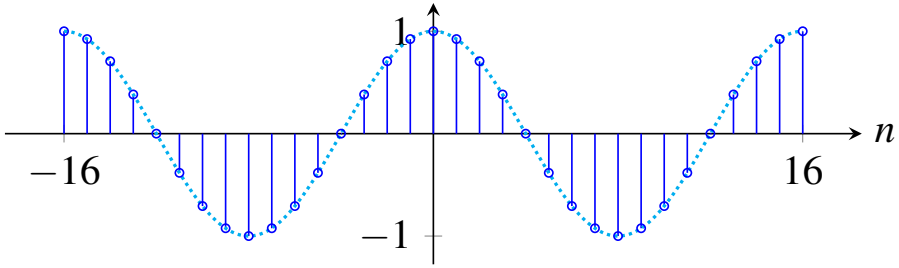
  where $0 \leq \Omega < 2\pi$.
- The rate of oscillation of x is least (i.e., x is constant) when $\Omega = 0$.
- The rate of oscillation of x is greatest when $\Omega = \pi$.
- As Ω increases from 0 to π, the rate of oscillation of x increases.
- As Ω increases from π to 2π, the rate of oscillation of x ***decreases***.

Effect of Increasing Frequency on Oscillation Rate

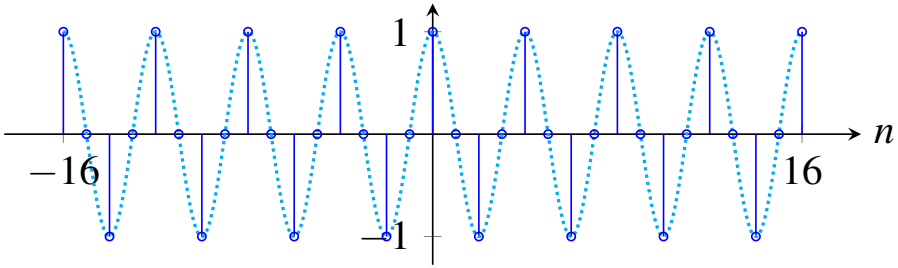
$\cos(0n) = \cos\left(\frac{0\pi}{8}n\right)$



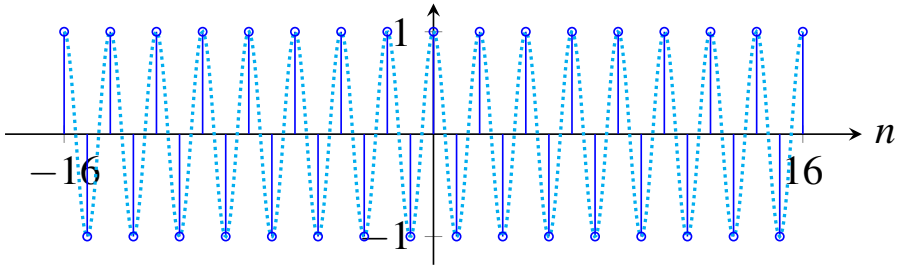
$\cos\left(\frac{\pi}{8}n\right)$



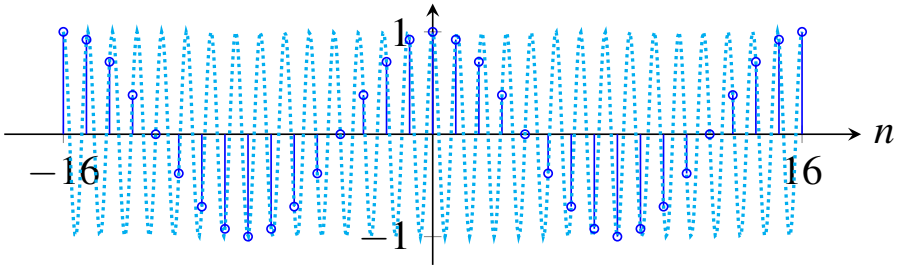
$\cos\left(\frac{\pi}{2}n\right) = \cos\left(\frac{4\pi}{8}n\right)$



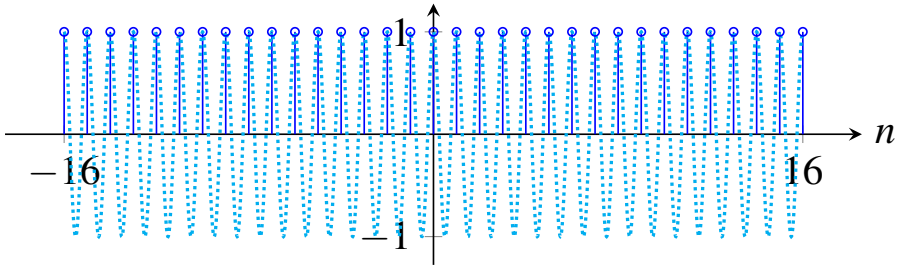
$\cos(\pi n) = \cos\left(\frac{8\pi}{8}n\right)$



$\cos\left(\frac{15\pi}{8}n\right)$



$\cos(2\pi n) = \cos\left(\frac{16\pi}{8}n\right)$



Complex Exponential Sequences

- A **complex exponential sequence** is a sequence of the form

$$x(n) = ca^n,$$

  where c and a are ***complex*** constants.
- Such a sequence can also be equivalently expressed in the form

$$x(n) = ce^{bn},$$

  where b is a ***complex*** constant chosen as $b = \ln a$. (This this form is more similar to that presented for CT complex exponentials).
- A complex exponential can exhibit one of a number of ***distinct modes of behavior***, depending on the values of the parameters c and a.
- For example, as special cases, complex exponentials include real exponentials and complex sinusoids.

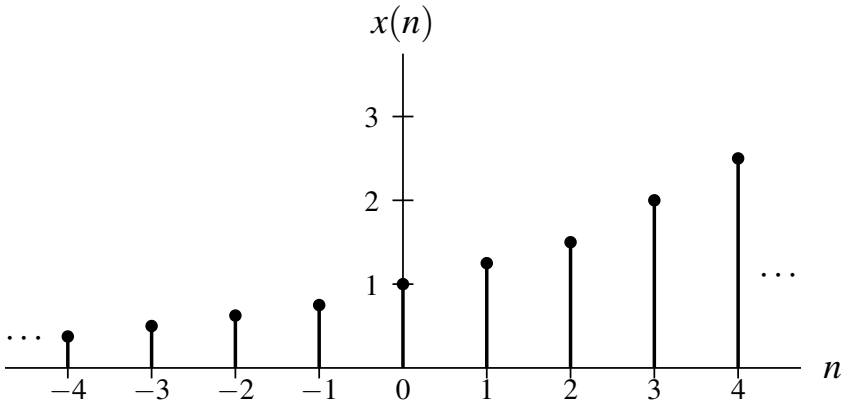
Real Exponential Sequences

- A **real exponential sequence** is a special case of a complex exponential

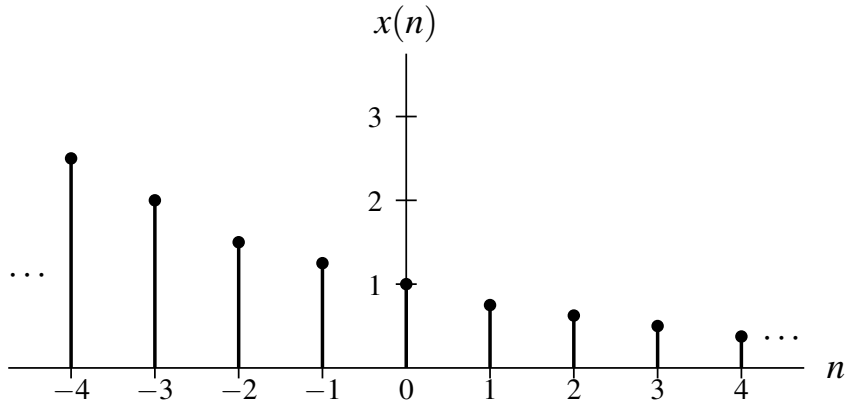
$$x(n) = ca^n,$$

  where c and a are restricted to be ***real*** numbers.
- A real exponential can exhibit one of ***several distinct modes*** of behavior, depending on the magnitude and sign of a.
- If $|a| > 1$, the magnitude of $x(n)$ ***increases*** exponentially as n increases (i.e., a growing exponential).
- If $|a| < 1$, the magnitude of $x(n)$ ***decreases*** exponentially as n increases (i.e., a decaying exponential).
- If $|a| = 1$, the magnitude of $x(n)$ is a ***constant***, independent of n.
- If $a > 0$, $x(n)$ has the ***same sign*** for all n.
- If $a < 0$, $x(n)$ ***alternates in sign*** as n increases/decreases.

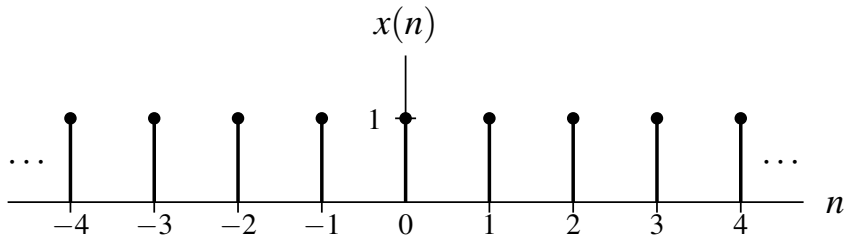
Real Exponential Sequences (Continued 1)



$|a| > 1, a > 0 \quad [a = \frac{5}{4}; c = 1]$

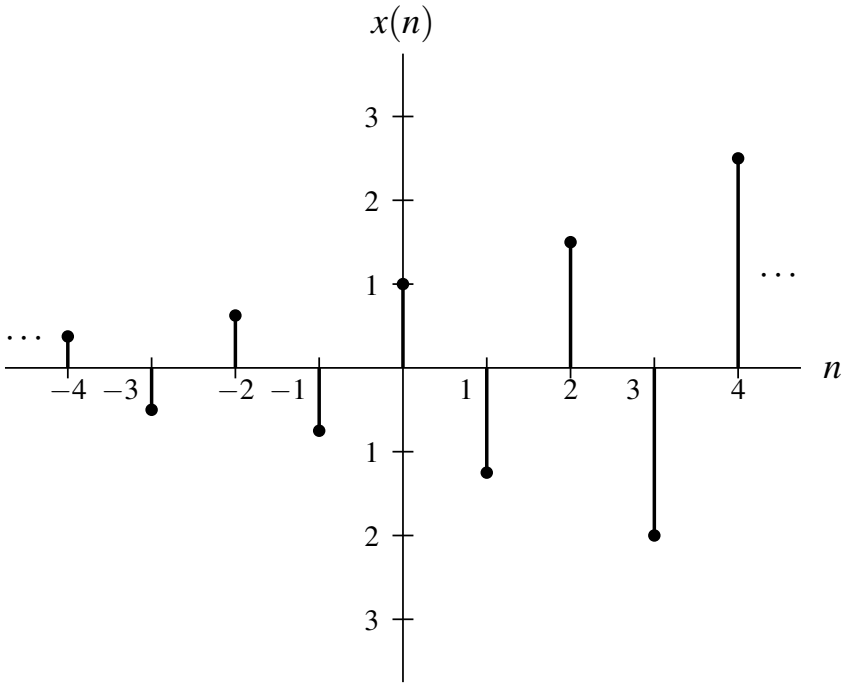


$|a| < 1, a > 0 \quad [a = \frac{4}{5}; c = 1]$

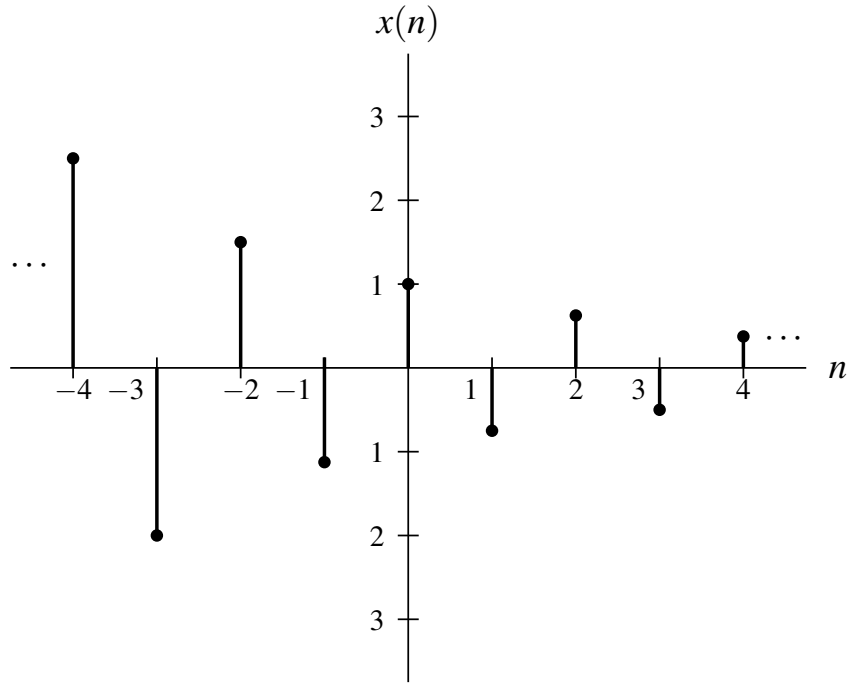


$|a| = 1, a > 0 \quad [a = 1; c = 1]$

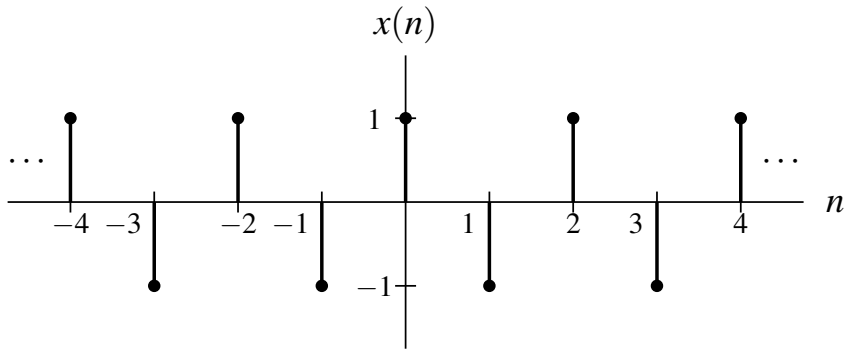
Real Exponential Sequences (Continued 2)



$|a| > 1, a < 0 \quad [a = -\frac{5}{4}; c = 1]$



$|a| < 1, a < 0 \quad [a = -\frac{4}{5}; c = 1]$



$|a| = 1, a < 0 \quad [a = -1; c = 1]$

Complex Sinusoidal Sequences

- A complex sinusoidal sequence is a special case of a complex exponential $x(n) = ca^n$, where c and a are ***complex*** and $|a| = 1$ (i.e., a is of the form $e^{j\Omega}$ where Ω is real).
- That is, a **complex sinusoidal sequence** is a sequence of the form

$$x(n) = ce^{j\Omega n},$$

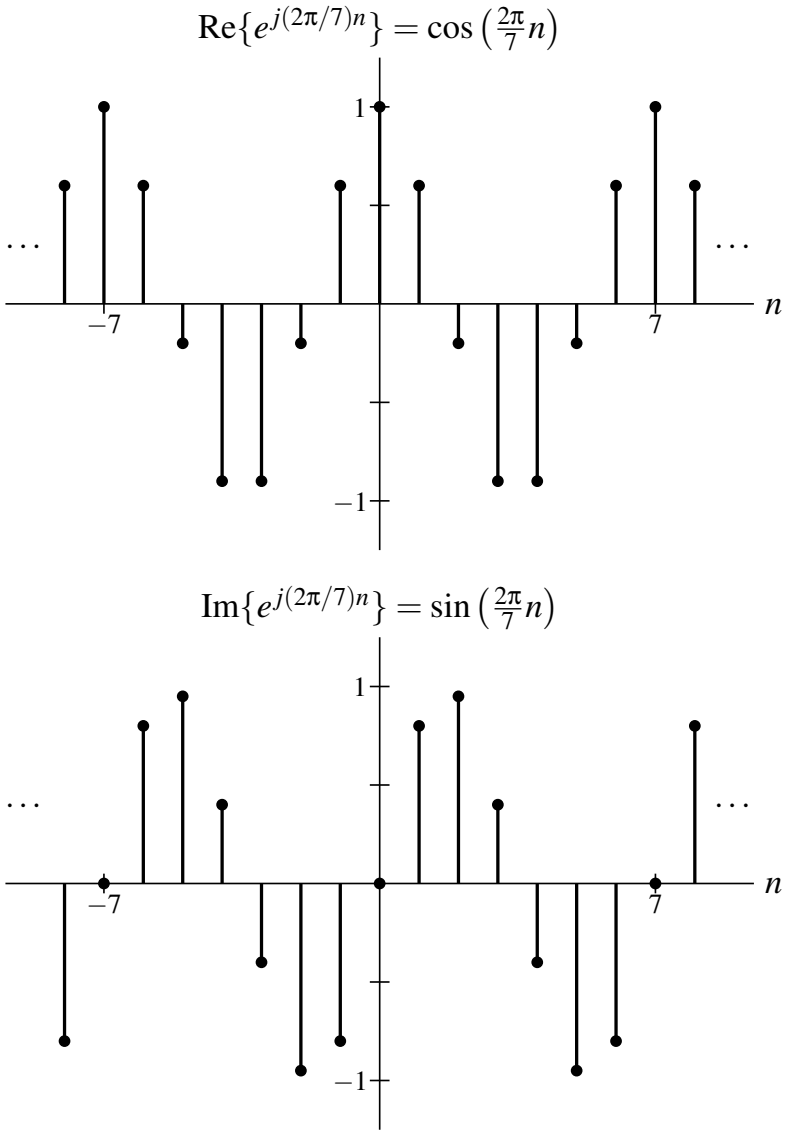
  where c is ***complex*** and Ω is ***real***.
- Using Euler's relation, we can rewrite $x(n)$ as

$$x(n) = \underbrace{|c|\cos(\Omega n + \arg c)}_{\mathrm{Re}\{x(n)\}} + j\underbrace{|c|\sin(\Omega n + \arg c)}_{\mathrm{Im}\{x(n)\}}.$$

- Thus, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are real sinusoids.
- A complex sinusoid is ***periodic*** if and only if $\frac{\Omega}{2\pi}$ is a ***rational number***, in which case the fundamental period is the ***smallest integer*** of the form $\frac{2\pi k}{|\Omega|}$ where k is a (strictly) positive integer.

Complex Sinusoidal Sequences (Continued)

- For $x(n) = e^{j(2\pi/7)n}$, the graphs of $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are shown below.



Oscillation Rate of Complex Sinusoidal Sequences

- Unlike their continuous-time counterparts, complex sinusoidal sequences have an upper bound on the rate at which they can oscillate.
- Since $x_k(n) = ce^{j(\Omega+2\pi k)n}$ is the same sequence for all integer k, we consider only $0 \leq \Omega < 2\pi$ without loss of generality.
- Consider the set of complex sinusoidal sequences of the form

$$x(n) = ce^{j\Omega n},$$

  where $0 \leq \Omega < 2\pi$.
- The rate of oscillation of x is least (i.e., x is constant) when $\Omega = 0$.
- The rate of oscillation of x is greatest when $\Omega = \pi$.
- As Ω increases from 0 to π, the rate of oscillation of x increases.
- As Ω increases from π to 2π, the rate of oscillation of x ***decreases***.

General Complex Exponential Sequences

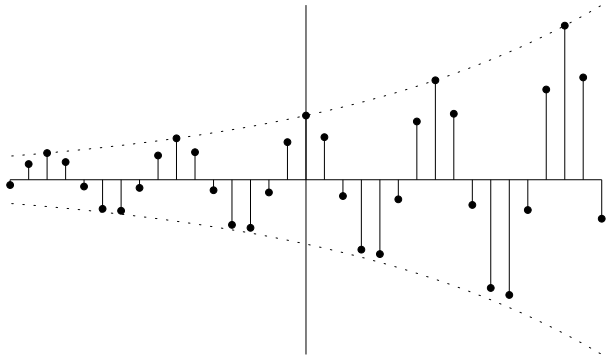
- In the most general case of a complex exponential sequence $x(n) = ca^n$, c and a are both ***complex***.
- Letting $c = |c|\,e^{j\theta}$ and $a = |a|\,e^{j\Omega}$ where θ and Ω are real, and using Euler's relation, we can rewrite $x(n)$ as

$$x(n) = \underbrace{|c|\,|a|^n \cos(\Omega n + \theta)}_{\mathrm{Re}\{x(n)\}} + j\underbrace{|c|\,|a|^n \sin(\Omega n + \theta)}_{\mathrm{Im}\{x(n)\}}.$$

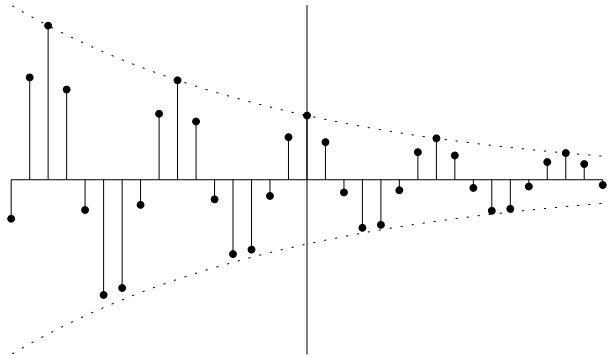
- Thus, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are each the product of a real exponential and real sinusoid.
- One of ***several distinct modes*** of behavior is exhibited by x, depending on the value of a.
- If $|a| = 1$, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are ***real sinusoids***.
- If $|a| > 1$, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are each the ***product of a real sinusoid and a growing real exponential***.
- If $|a| < 1$, $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are each the ***product of a real sinusoid and a decaying real exponential***.

General Complex Exponential Sequences (Continued)

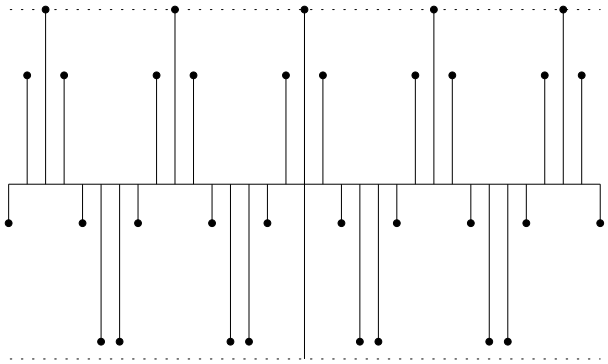
- The ***various modes of behavior*** for $\mathrm{Re}\{x\}$ and $\mathrm{Im}\{x\}$ are illustrated below.



$|a| > 1$



$|a| < 1$



$|a| = 1$

Relationship Between Complex Exponentials and Real Sinusoids

- From Euler's relation, a complex sinusoid can be expressed as the sum of two real sinusoids as

$$ce^{j\Omega n} = c\cos(\Omega n) + jc\sin(\Omega n).$$

- Moreover, a real sinusoid can be expressed as the sum of two complex sinusoids using the identities

$$c\cos(\Omega n + \theta) = \frac{c}{2}\left[e^{j(\Omega n+\theta)} + e^{-j(\Omega n+\theta)}\right] \quad \text{and}$$

$$c\sin(\Omega n + \theta) = \frac{c}{2j}\left[e^{j(\Omega n+\theta)} - e^{-j(\Omega n+\theta)}\right].$$

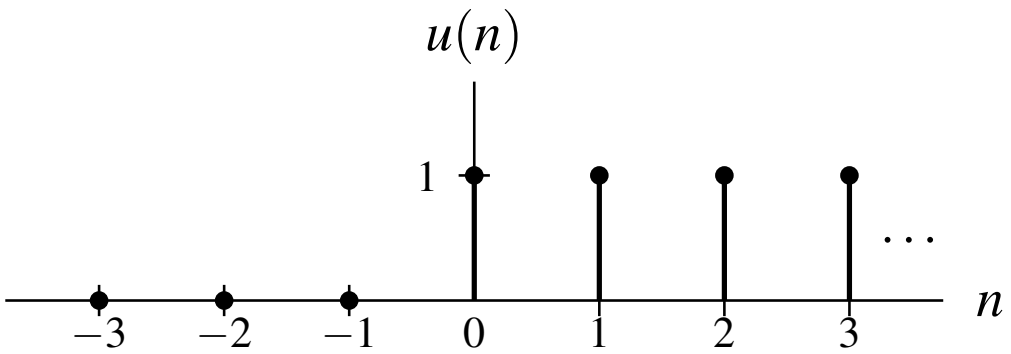
- Note that, above, we are simply ***restating results*** from the (appendix) material on complex analysis.

Unit-Step Sequence

- The **unit-step sequence**, denoted u, is defined as

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

- A plot of this sequence is shown below.



Unit Rectangular Pulses

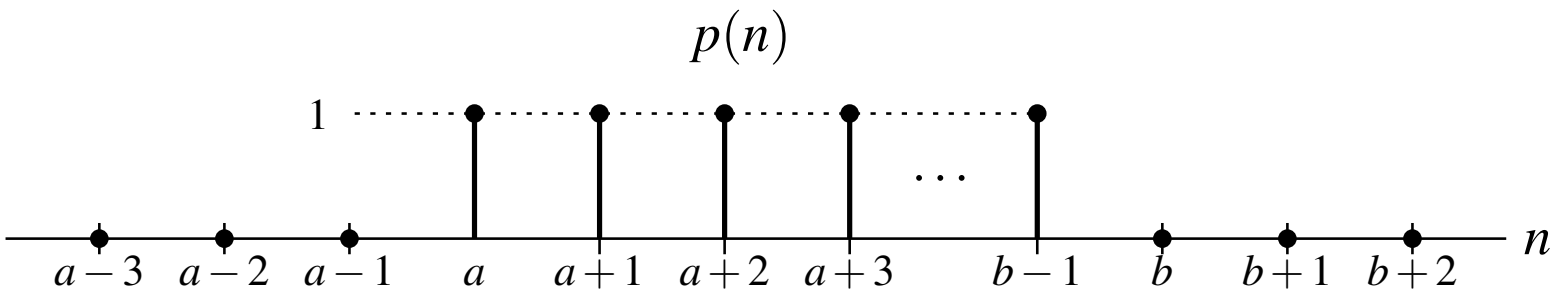
- A **unit rectangular pulse** is a sequence of the form

$$p(n) = \begin{cases} 1 & a \leq n < b \\ 0 & \text{otherwise} \end{cases}$$

  where a and b are integer constants satisfying $a < b$.
- Such a sequence can be expressed in terms of the unit-step sequence as

$$p(n) = u(n-a) - u(n-b).$$

- The graph of a unit rectangular pulse has the general form shown below.



Unit-Impulse Sequence

- The **unit-impulse sequence** (also known as the **delta sequence**), denoted δ, is defined as

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

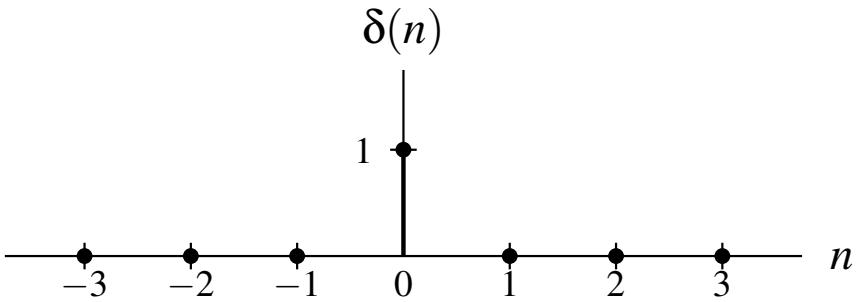
- The first-order difference of u is δ. That is,

$$\delta(n) = u(n) - u(n-1).$$

- The running sum of δ is u. That is,

$$u(n) = \sum_{k=-\infty}^{n} \delta(k).$$

- A plot of δ is shown below.



Properties of the Unit-Impulse Sequence

- For any sequence x and any integer constant n_0, the following identity holds:

$$x(n)\delta(n-n_0) = x(n_0)\delta(n-n_0).$$

- For any sequence x and any integer constant n_0, the following identity holds:

$$\sum_{n=-\infty}^{\infty} x(n)\delta(n-n_0) = x(n_0).$$

- Trivially, the sequence δ is also even.

Representing Rectangular Pulses (Using Unit-Step Sequences)

- For integer constants a and b where $a < b$, consider a sequence x of the form

$$x(n) = \begin{cases} 1 & a \leq n < b \\ 0 & \text{otherwise} \end{cases}$$

  (i.e., x is a ***rectangular pulse*** of height one that is nonzero from a to $b-1$ inclusive).

- The sequence x can be equivalently written as

$$x(n) = u(n-a) - u(n-b)$$

  (i.e., the difference of two time-shifted unit-step sequences).

- Unlike the original expression for x, this latter expression for x ***does not involve multiple cases***.

- In effect, by using unit-step sequences, we have collapsed a formula involving multiple cases into a single expression.

Representing Sequences Using Unit-Step Sequences

- The idea from the previous slide can be extended to handle any sequence that is defined in a ***piecewise manner*** (i.e., via an expression involving multiple cases).
- That is, by using unit-step sequences, we can always collapse a formula involving multiple cases into a single expression.
- Often, simplifying a formula in this way can be quite beneficial.

Section 8.4

Discrete-Time (DT) Systems

DT Systems

- A system with input x and output y can be described by the equation

$$y = \mathcal{H}x,$$

  where $\mathcal{H}$ denotes an operator (i.e., transformation).
- Note that the operator $\mathcal{H}$ ***maps a sequence to a sequence*** (not a number to a number).
- Alternatively, we can express the above relationship using the notation

$$x \overset{\mathcal{H}}{\longrightarrow} y.$$

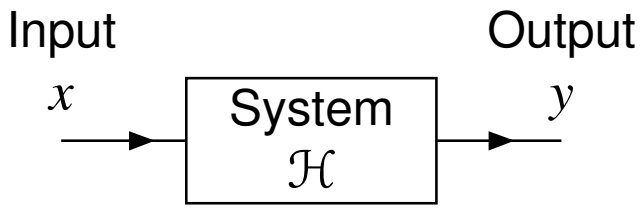
- If clear from the context, the operator $\mathcal{H}$ is often omitted, yielding the abbreviated notation

$$x \rightarrow y.$$

- Note that the symbols "$\rightarrow$" and "=" have ***very different*** meanings.
- The symbol "$\rightarrow$" should be read as "***produces***" (not as "equals").

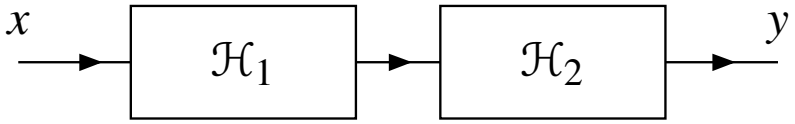
Block Diagram Representations

- Often, a system defined by the operator $\mathcal{H}$ and having the input x and output y is represented in the form of a ***block diagram*** as shown below.

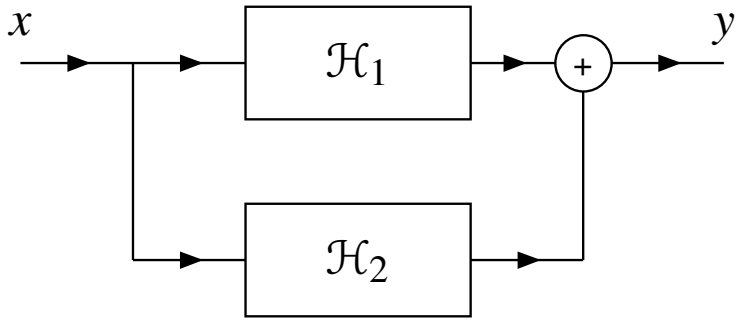


Interconnection of Systems

- ***Two basic ways*** in which systems can be interconnected are shown below.



Series



Parallel

- A **series** (or **cascade**) connection ties the output of one system to the input of the other.
- The overall series-connected system is described by the equation

$$y = \mathcal{H}_2\mathcal{H}_1 x.$$

- A **parallel** connection ties the inputs of both systems together and sums their outputs.
- The overall parallel-connected system is described by the equation

$$y = \mathcal{H}_1 x + \mathcal{H}_2 x.$$

Section 8.5

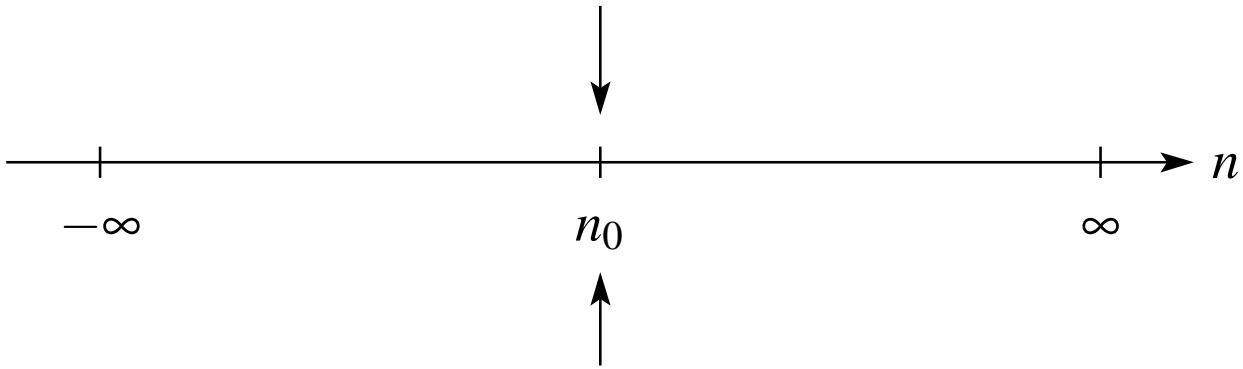
Properties of (DT) Systems

Memory

- A system $\mathcal{H}$ is said to be **memoryless** if, for every integer constant n_0, $\mathcal{H}x(n_0)$ does not depend on $x(n)$ for some $n \neq n_0$.
- In other words, a memoryless system is such that the value of its output at any given point in time can depend on the value of its input at only the ***same*** point in time.
- A system that is not memoryless is said to have **memory**.
- Although simple, a memoryless system is ***not very flexible***, since its current output value cannot rely on past or future values of the input.

Memory (Continued)

If the system $\mathcal{H}$ is memoryless,
the output $\mathcal{H}x$ at n_0
can depend on the input x
only at n_0.



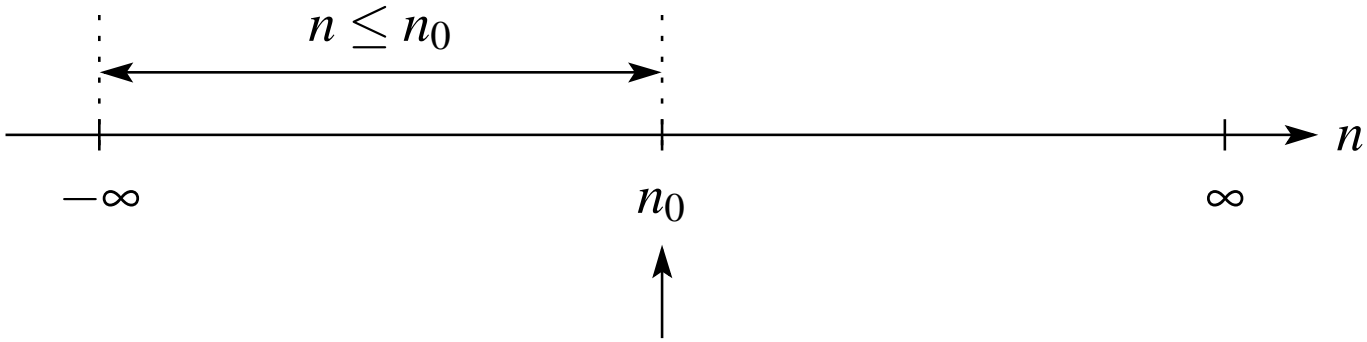
Consider the calculation of the
output $\mathcal{H}x$ at n_0.

Causality

- A system $\mathcal{H}$ is said to be **causal** if, for every integer constant n_0, $\mathcal{H}x(n_0)$ does not depend on $x(n)$ for some $n > n_0$.
- In other words, a causal system is such that the value of its output at any given point in time can depend on the value of its input at only the ***same or earlier points*** in time (i.e., ***not later points in time***).
- If the independent variable n represents time, a system must be causal in order to be ***physically realizable***.
- Noncausal systems can sometimes be useful in practice, however, since the independent variable ***need not always represent time*** (e.g., the independent variable might represent position).
- A memoryless system is always causal, although the converse is not necessarily true.

Causality (Continued)

If the system $\mathcal{H}$ is causal,
the output $\mathcal{H}x$ at n_0
can depend on the input x
only at points $n \leq n_0$.



Consider the calculation of the
output $\mathcal{H}x$ at n_0.

Invertibility

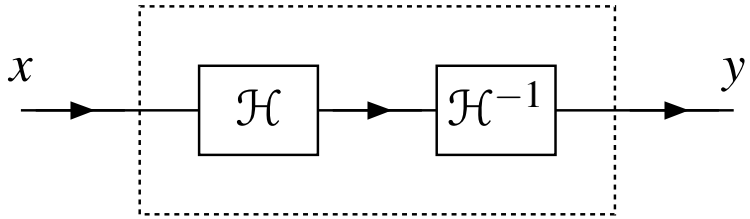
- The **inverse** of a system $\mathcal{H}$ is another system $\mathcal{H}^{-1}$ such that, for every sequence x,

$$\mathcal{H}^{-1}\mathcal{H}x = x$$

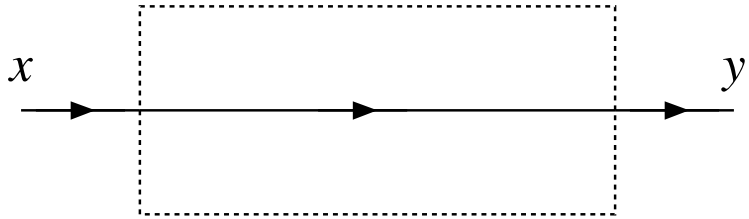
  (i.e., the system formed by the cascade interconnection of $\mathcal{H}$ followed by $\mathcal{H}^{-1}$ is a system whose input and output are equal).
- A system is said to be **invertible** if it has a corresponding inverse system (i.e., its inverse exists).
- Equivalently, a system is invertible if its input x can always be ***uniquely*** determined from its output y.
- An invertible system will always produce ***distinct outputs*** from any two ***distinct inputs***.
- To show that a system is ***invertible***, we simply find the ***inverse system***.
- To show that a system is ***not invertible***, we find ***two distinct inputs*** that result in ***identical outputs***.
- In practical terms, invertible systems are "nice" in the sense that their ***effects can be undone***.

Invertibility (Continued)

- A system $\mathcal{H}^{-1}$ being the inverse of $\mathcal{H}$ means that the following two systems are equivalent (i.e., $\mathcal{H}^{-1}\mathcal{H}$ ***is an identity***):



System 1: $y = \mathcal{H}^{-1}\mathcal{H}x$



System 2: $y = x$

Bounded-Input Bounded-Output (BIBO) Stability

- A system $\mathcal{H}$ is **BIBO stable** if, for every bounded sequence x, $\mathcal{H}x$ is bounded (i.e., $|x(n)| < \infty$ for all n implies that $|\mathcal{H}x(n)| < \infty$ for all n).
- In other words, a BIBO stable system is such that it guarantees to always produce a bounded output as long as its input is bounded.
- To show that a system is ***BIBO stable***, we must show that ***every bounded input*** leads to a ***bounded output***.
- To show that a system is ***not BIBO stable***, we need only find a single ***bounded input*** that leads to an ***unbounded output***.
- In practical terms, a BIBO stable system is ***well behaved*** in the sense that, as long as the system input remains finite for all time, the output will also remain finite for all time.
- Usually, a system that is not BIBO stable will have ***serious safety issues***.
- For example, a portable music player with a battery input of 3.7 volts and headset output of ∞ volts would result in one vaporized human (and likely a big lawsuit as well).

Time Invariance (TI)

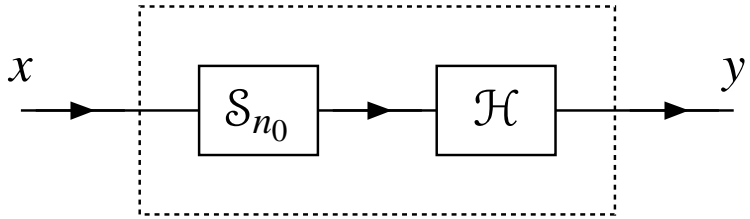
- A system $\mathcal{H}$ is said to be **time invariant (TI)** (or **shift invariant (SI)**) if, for every sequence x and every integer n_0, the following condition holds:

$$\mathcal{H}x(n-n_0) = \mathcal{H}x'(n) \text{ for all } n, \quad \text{where} \quad x'(n) = x(n-n_0)$$

  (i.e., $\mathcal{H}$ ***commutes with time shifts***).
- In other words, a system is time invariant if a time shift (i.e., advance or delay) in the input always results only in an ***identical time shift*** in the output.
- A system that is not time invariant is said to be **time varying**.
- In simple terms, a time invariant system is a system whose behavior ***does not change*** with respect to time.
- Practically speaking, compared to time-varying systems, time-invariant systems are much ***easier to design and analyze***, since their behavior does not change with respect to time.

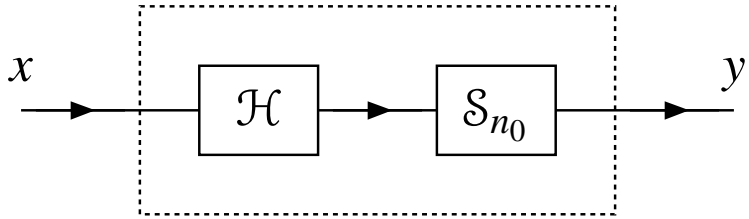
Time Invariance (Continued)

- Let $\mathcal{S}_{n_0}$ denote an operator that applies a ***time shift of*** n_0 to a sequence (i.e., $\mathcal{S}_{n_0}x(n) = x(n-n_0)$).
- A system $\mathcal{H}$ is ***time invariant*** if and only if the following two systems are equivalent (i.e., $\mathcal{H}$ ***commutes with*** $\mathcal{S}_{n_0}$):



System 1: $y = \mathcal{H}\mathcal{S}_{n_0}x$

$$\begin{bmatrix} y(n) = \mathcal{H}x'(n) \\ x'(n) = \mathcal{S}_{n_0}x(n) = x(n-n_0) \end{bmatrix}$$



System 2: $y = \mathcal{S}_{n_0}\mathcal{H}x$

$$\left[y(n) = \mathcal{H}x(n-n_0)\right]$$

Additivity, Homogeneity, and Linearity

- A system $\mathcal{H}$ is said to be **additive** if, for all sequences x_1 and x_2, the following condition holds:

$$\mathcal{H}(x_1 + x_2) = \mathcal{H}x_1 + \mathcal{H}x_2$$

  (i.e., $\mathcal{H}$ ***commutes with sums***).
- A system $\mathcal{H}$ is said to be **homogeneous** if, for every sequence x and every complex constant a, the following condition holds:

$$\mathcal{H}(ax) = a\mathcal{H}x$$

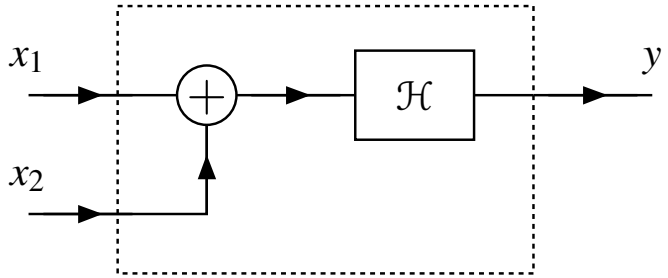
  (i.e., $\mathcal{H}$ ***commutes with multiplication by a constant***).
- A system that is both additive and homogeneous is said to be **linear**.
- In other words, a system $\mathcal{H}$ is ***linear***, if for all sequences x_1 and x_2 and all complex constants a_1 and a_2, the following condition holds:

$$\mathcal{H}(a_1x_1 + a_2x_2) = a_1\mathcal{H}x_1 + a_2\mathcal{H}x_2$$

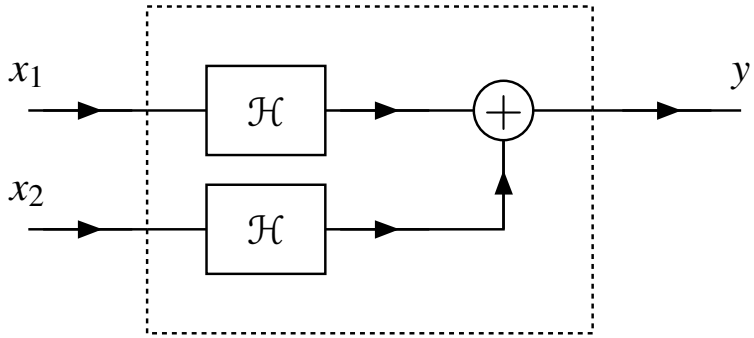
  (i.e., $\mathcal{H}$ ***commutes with linear combinations***).
- The linearity property is also referred to as the **superposition** property.
- Practically speaking, linear systems are much ***easier to design and analyze*** than nonlinear systems.

Additivity, Homogeneity, and Linearity (Continued 1)

- The system $\mathcal{H}$ is ***additive*** if and only if the following two systems are equivalent (i.e., ***$\mathcal{H}$ commutes with addition***):

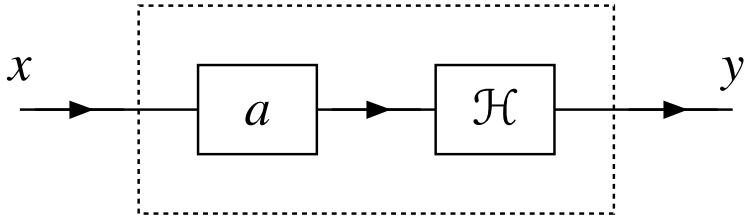


System 1: $y = \mathcal{H}(x_1 + x_2)$

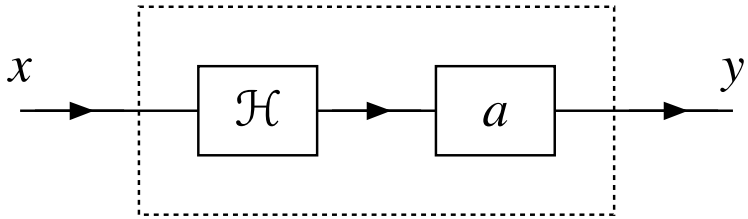


System 2: $y = \mathcal{H}x_1 + \mathcal{H}x_2$

- The system $\mathcal{H}$ is ***homogeneous*** if and only if the following two systems are equivalent (i.e., ***$\mathcal{H}$ commutes with scalar multiplication***):



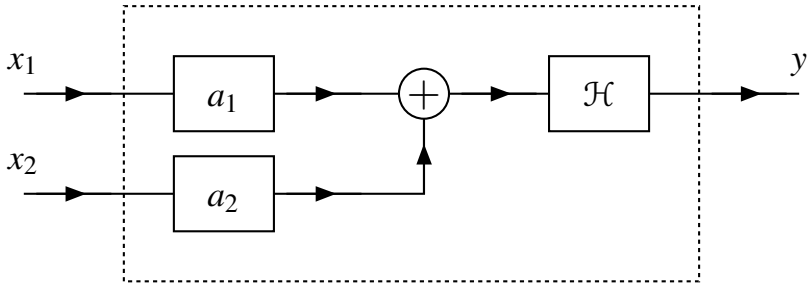
System 1: $y = \mathcal{H}(ax)$



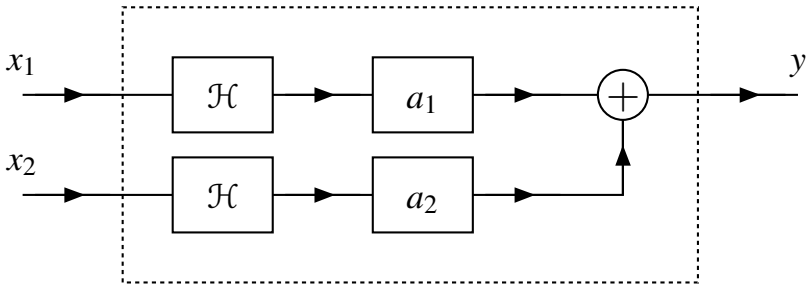
System 2: $y = a\mathcal{H}x$

Additivity, Homogeneity, and Linearity (Continued 2)

- The system $\mathcal{H}$ is ***linear*** if and only if the following two systems are equivalent (i.e., ***$\mathcal{H}$ commutes with linear combinations***):



System 1: $y = \mathcal{H}(a_1 x_1 + a_2 x_2)$



System 2: $y = a_1 \mathcal{H} x_1 + a_2 \mathcal{H} x_2$

Eigensequences of Systems

- A sequence x is said to be an **eigensequence** of the system $\mathcal{H}$ with the **eigenvalue** λ if

$$\mathcal{H}x = \lambda x,$$

  where λ is a complex constant.
- In other words, the system $\mathcal{H}$ acts as an ideal amplifier for each of its eigensequences x, where the amplifier gain is given by the corresponding eigenvalue λ.
- Different systems have different eigensequences.
- Many of the mathematical tools developed for the study of DT systems have eigensequences as their basis.

Part 9

Discrete-Time Linear Time-Invariant (LTI) Systems

Why Linear Time-Invariant (LTI) Systems?

- In engineering, linear time-invariant (LTI) systems play a very important role.
- Very powerful mathematical tools have been developed for analyzing LTI systems.
- LTI systems are much easier to analyze than systems that are not LTI.
- In practice, systems that are not LTI can be well approximated using LTI models.
- So, even when dealing with systems that are not LTI, LTI systems still play an important role.

Section 9.1

Convolution

DT Convolution

- The (DT) **convolution** of the sequences x and h, denoted $x * h$, is defined as the sequence

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

- The convolution $x * h$ evaluated at the point n is simply a weighted sum of elements of x, where the weighting is given by h time reversed and shifted by n.
- Herein, the asterisk symbol (i.e., "$*$") will always be used to denote convolution, not multiplication.
- As we shall see, convolution is used extensively in the theory of (DT) systems.
- In particular, convolution has a special significance in the context of (DT) LTI systems.

Practical Convolution Computation

- To compute the convolution

$$x * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k),$$

we proceed as follows:

1. Plot $x(k)$ and $h(n-k)$ as a function of k.
2. Initially, consider an arbitrarily large negative value for n. This will result in $h(n-k)$ being shifted very far to the left on the time axis.
3. Write the mathematical expression for $x * h(n)$.
4. Increase n gradually until the expression for $x * h(n)$ changes form. Record the interval over which the expression for $x * h(n)$ was valid.
5. Repeat steps 3 and 4 until n is an arbitrarily large positive value. This corresponds to $h(n-k)$ being shifted very far to the right on the time axis.
6. The results for the various intervals can be combined in order to obtain an expression for $x * h(n)$ for all n.

Properties of Convolution

- The convolution operation is ***commutative***. That is, for any two sequences x and h,

$$x * h = h * x.$$

- The convolution operation is ***associative***. That is, for any sequences x, h_1, and h_2,

$$(x * h_1) * h_2 = x * (h_1 * h_2).$$

- The convolution operation is ***distributive*** with respect to addition. That is, for any sequences x, h_1, and h_2,

$$x * (h_1 + h_2) = x * h_1 + x * h_2.$$

Representation of Sequences Using Impulses

- For any sequence x,

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

- Thus, any sequence x can be written in terms of an expression involving δ.
- Moreover, δ is the ***convolutional identity***. That is, for any sequence x,

$$x * \delta = x.$$

Circular Convolution

- The convolution of two periodic sequences is usually not well defined.
- This motivates an alternative notion of convolution for periodic sequences known as circular convolution.
- The **circular convolution** (also known as the DT periodic convolution) of the N-periodic sequences x and h, denoted $x \circledast h$, is defined as

$$x \circledast h(n) = \sum_{k=\langle N \rangle} x(k)h(n-k) = \sum_{k=0}^{N-1} x(k)h(\mathrm{mod}(n-k,N)),$$

  where $\mathrm{mod}(a,b)$ is the remainder after division when a is divided by b.
- The circular convolution and (linear) convolution of the N-periodic sequences x and h are related as follows:

$$x \circledast h(n) = x_0 * h(n) \quad \text{where} \quad x(n) = \sum_{k=-\infty}^{\infty} x_0(n-kN)$$

  (i.e., $x_0(n)$ equals $x(n)$ over a single period of x and is zero elsewhere).

Section 9.2

Convolution and LTI Systems

Impulse Response

- The response h of a system $\mathcal{H}$ to the input δ is called the **impulse response** of the system (i.e., $h = \mathcal{H}\delta$).
- For any LTI system with input x, output y, and impulse response h, the following relationship holds:

$$y = x * h.$$

- In other words, a LTI system simply ***computes a convolution***.
- Furthermore, a LTI system is ***completely characterized*** by its impulse response.
- That is, if the impulse response of a LTI system is known, we can determine the response of the system to any input.

Step Response

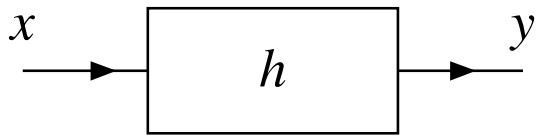
- The response s of a system $\mathcal{H}$ to the input u is called the **step response** of the system (i.e., $s = \mathcal{H}u$).
- The impulse response h and step response s of a system are related as

$$h(n) = s(n) - s(n-1).$$

- Therefore, the impulse response of a system can be determined from its step response by (first-order) differencing.

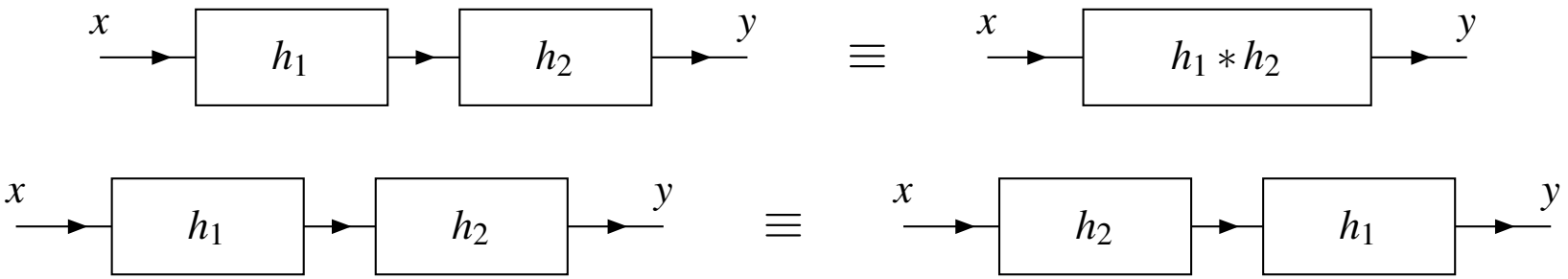
Block Diagram of LTI Systems

- Often, it is convenient to represent a (DT) LTI system in block diagram form.
- Since such systems are completely characterized by their impulse response, we often label a system with its impulse response.
- That is, we represent a system with input x, output y, and impulse response h, as shown below.

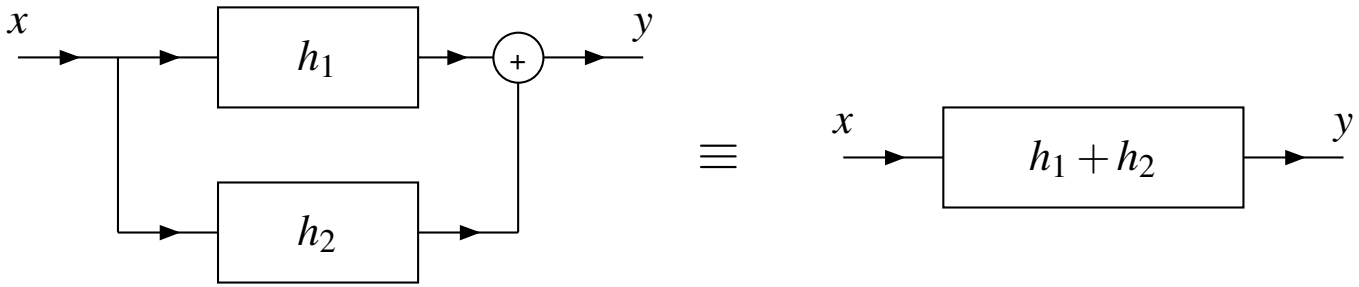


Interconnection of LTI Systems

- The ***series*** interconnection of the LTI systems with impulse responses h_1 and h_2 is the LTI system with impulse response $h = h_1 * h_2$. That is, we have the equivalences shown below.



- The ***parallel*** interconnection of the LTI systems with impulse responses h_1 and h_2 is a LTI system with the impulse response $h = h_1 + h_2$. That is, we have the equivalence shown below.



Section 9.3

Properties of LTI Systems

Memory

- A LTI system with impulse response h is memoryless if and only if

$$h(n) = 0 \quad \text{for all } n \neq 0.$$

- That is, a LTI system is memoryless if and only if its impulse response h is of the form

$$h(n) = K\delta(n),$$

where K is a complex constant.

- Consequently, every memoryless LTI system with input x and output y is characterized by an equation of the form

$$y = x * (K\delta) = Kx$$

(i.e., the system is an ideal amplifier).

- For a LTI system, the memoryless constraint is extremely restrictive (as every memoryless LTI system is an ideal amplifier).

Causality

- A LTI system with impulse response h is causal if and only if

$$h(n) = 0 \quad \text{for all } n < 0$$

(i.e., h is a causal sequence).

- It is due to the above relationship that we call a sequence x, satisfying

$$x(n) = 0 \quad \text{for all } n < 0,$$

a causal sequence.

Invertibility

- The inverse of a LTI system, if such a system exists, is a LTI system.
- Let h and h_{inv} denote the impulse responses of a LTI system and its (LTI) inverse, respectively. Then,

$$h * h_{\text{inv}} = \delta.$$

- Consequently, a LTI system with impulse response h is invertible if and only if there exists a sequence h_{inv} such that

$$h * h_{\text{inv}} = \delta.$$

- Except in simple cases, the above condition is often quite difficult to test.

BIBO Stability

- A LTI system with impulse response h is BIBO stable if and only if

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

(i.e., h is ***absolutely summable***).

Eigensequences of LTI Systems

- As it turns out, every complex exponential is an eigensequence of all LTI systems.
- For a LTI system $\mathcal{H}$ with impulse response h,

$$\mathcal{H}\{z^n\}(n) = H(z)z^n,$$

where z is a complex constant and

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}.$$

- That is, z^n is an eigensequence of a LTI system and $H(z)$ is the corresponding eigenvalue.
- We refer to H as the **system function** (or **transfer function**) of the system $\mathcal{H}$.
- From above, we can see that the response of a LTI system to a complex exponential is the same complex exponential multiplied by the complex factor $H(z)$.

Representation of Sequences Using Eigensequences

- Consider a LTI system with input x, output y, and system function H.
- Suppose that the input x can be expressed as the linear combination of complex exponentials

$$x(n) = \sum_k a_k z_k^n,$$

where the a_k and z_k are complex constants.

- Using the fact that complex exponentials are eigenfunctions of LTI systems, we can conclude

$$y(n) = \sum_k a_k H(z_k) z_k^n.$$

- Thus, if an input to a LTI system can be expressed as a linear combination of complex exponentials, the output can also be expressed as linear combination of the ***same*** complex exponentials.
- The above formula can be used to determine the output of a LTI system from its input in a way that does not require convolution.

Part 10

Discrete-Time Fourier Series (DTFS)

Introduction

- The Fourier series is a representation for ***periodic*** sequences.
- With a Fourier series, a sequence is represented as a ***linear combination of complex sinusoids***.
- The use of complex sinusoids is desirable due to their numerous attractive properties.
- Perhaps, most importantly, complex sinusoids are ***eigensequences*** of (DT) LTI systems.

Section 10.1

Fourier Series

Harmonically-Related Complex Sinusoids

- A set of periodic complex sinusoids is said to be **harmonically related** if there exists some constant $\frac{2\pi}{N}$ such that the fundamental frequency of each complex sinusoid is an integer multiple of $\frac{2\pi}{N}$.
- Consider the set of harmonically-related complex sinusoids given by

$$\phi_k(n) = e^{j(2\pi/N)kn} \quad \text{for all integer } k.$$

- In the above set $\{\phi_k\}$, only N elements are distinct, since

$$\phi_k = \phi_{k+N} \quad \text{for all integer } k.$$

- Since the fundamental frequency of each of the harmonically-related complex sinusoids is an integer multiple of $\frac{2\pi}{N}$, a linear combination of these complex sinusoids must be N-periodic.

DT Fourier Series (DTFS)

- An N-periodic complex-valued sequence x can be represented as a linear combination of harmonically-related complex sinusoids as

$$x(n) = \sum_{k=\langle N \rangle} a_k e^{j(2\pi/N)kn},$$

  where $\sum_{k=\langle N \rangle}$ denotes summation over any N consecutive integers (e.g., $[0\,..\,N-1]$). (The summation can be taken over any N consecutive integers, due to the N-periodic nature of x and $e^{j(2\pi/N)kn}$.)
- The above representation of x is known as the (DT) **Fourier series** and the a_k are called **Fourier series coefficients**.
- The above formula for x is often called the **Fourier series synthesis equation**.
- To denote that the sequence x has the Fourier series coefficient sequence a, we write

$$x(n) \overset{\text{DTFS}}{\longleftrightarrow} a_k.$$

DT Fourier Series (DTFS) (Continued)

- A periodic sequence x with fundamental period N has the Fourier series coefficient sequence a given by

$$a_k = \frac{1}{N} \sum_{n=\langle N \rangle} x(n) e^{-j(2\pi/N)kn}.$$

  (The summation can be taken over any N consecutive integers due to the N-periodic nature of x and $e^{-j(2\pi/N)kn}$.)

- The above equation for a_k is often referred to as the **Fourier series analysis equation**.
- Due to the N-periodic nature of x and $e^{-j(2\pi/N)kn}$, the sequence a is also N-periodic.

Convergence of Fourier Series

- Since the analysis and synthesis equations for (DT) Fourier series involve only ***finite*** sums (as opposed to infinite series), convergence is not a significant issue of concern.
- If an N-periodic sequence is bounded (i.e., is finite in value), its Fourier series coefficient sequence will exist and be bounded and the Fourier series analysis and synthesis equations must converge.

Section 10.2

Properties of Fourier Series

Properties of (DT) Fourier Series

$$x(n) \overset{\text{DTFS}}{\longleftrightarrow} a_k \quad \text{and} \quad y(n) \overset{\text{DTFS}}{\longleftrightarrow} b_k$$

Property	Time Domain	Fourier Domain
Linearity	$\alpha x(n) + \beta y(n)$	$\alpha a_k + \beta b_k$
Translation	$x(n - n_0)$	$e^{-jk(2\pi/N)n_0} a_k$
Modulation	$e^{j(2\pi/N)k_0 n} x(n)$	a_{k-k_0}
Reflection	$x(-n)$	a_{-k}
Conjugation	$x^*(n)$	a^*_{-k}
Duality	a_n	$\frac{1}{N} x(-k)$
Periodic Convolution	$x \circledast y(n)$	$N a_k b_k$
Multiplication	$x(n)y(n)$	$a \circledast b_k$

Property	
Parseval's Relation	$\frac{1}{N} \sum_{n=\langle N \rangle} \lvert x(n) \rvert^2 = \sum_{k=\langle N \rangle} \lvert a_k \rvert^2$
Even Symmetry	x is even $\Leftrightarrow$ a is even
Odd Symmetry	x is odd $\Leftrightarrow$ a is odd
Real / Conjugate Symmetry	x is real $\Leftrightarrow$ a is conjugate symmetric

Linearity

- Let x and y be N-periodic sequences. If $x(n) \overset{\text{DTFS}}{\longleftrightarrow} a_k$ and $y(n) \overset{\text{DTFS}}{\longleftrightarrow} b_k$, then

$$\alpha x(n) + \beta y(n) \overset{\text{DTFS}}{\longleftrightarrow} \alpha a_k + \beta b_k,$$

  where α and β are complex constants.
- That is, a linear combination of sequences produces the same linear combination of their Fourier series coefficients.

Translation (Time Shifting)

- Let x denote a periodic sequence with period N. If $x(n) \overset{\text{DTFS}}{\longleftrightarrow} c_k$, then

$$x(n-n_0) \overset{\text{DTFS}}{\longleftrightarrow} e^{-jk(2\pi/N)n_0} c_k,$$

  where n_0 is an integer constant.
- In other words, time shifting a periodic sequence changes the argument (but not magnitude) of its Fourier series coefficients.

Modulation (Frequency Shifting)

- Let x denote a periodic sequence with period N. If $x(n) \overset{\text{DTFS}}{\longleftrightarrow} c_k$, then

$$e^{j(2\pi/N)k_0 n} x(n) \overset{\text{DTFS}}{\longleftrightarrow} c_{k-k_0},$$

  where k_0 is an integer constant.
- That is, multiplying a sequence by a complex sinusoid whose frequency is an integer multiple of $\frac{2\pi}{N}$ results in a translation of the corresponding Fourier series coefficient sequence.

Reflection (Time Reversal)

- Let x denote a periodic sequence with period N. If $x(n) \overset{\text{DTFS}}{\longleftrightarrow} c_k$, then

$$x(-n) \overset{\text{DTFS}}{\longleftrightarrow} c_{-k}.$$

- That is, time reversing a sequence results in a time reversal of the corresponding Fourier series coefficient sequence.

Conjugation

- Let x denote a periodic sequence with period N. If $x(n) \overset{\text{DTFS}}{\longleftrightarrow} c_k$, then

$$x^*(n) \overset{\text{DTFS}}{\longleftrightarrow} c^*_{-k}.$$

- In other words, conjugating a sequence has the effect of time reversing and conjugating the corresponding Fourier series coefficient sequence.

- Let x denote a periodic sequence with period N. If $x(n) \overset{\text{DTFS}}{\longleftrightarrow} a(k)$, then

$$a(n) \overset{\text{DTFS}}{\longleftrightarrow} \frac{1}{N} x(-k).$$

- This is known as the **duality property** of Fourier series.
- This property follows from the high degree of symmetry in the analysis and synthesis Fourier-series equations, which are respectively given by

$$x(m) = \sum_{\ell=\langle N \rangle} a(\ell) e^{j(2\pi/N)\ell m} \quad \text{and} \quad a(m) = \frac{1}{N} \sum_{\ell=\langle N \rangle} x(\ell) e^{-j(2\pi/N)m\ell}.$$

- That is, the analysis and synthesis equations are identical except for a ***factor of N*** and ***different sign*** in the parameter for the exponential function.

Periodic Convolution

- Let x and y be N-periodic sequences. If $x(n) \overset{\text{DTFS}}{\longleftrightarrow} a_k$ and $y(n) \overset{\text{DTFS}}{\longleftrightarrow} b_k$, then

$$x \circledast y(n) \overset{\text{DTFS}}{\longleftrightarrow} N a_k b_k.$$

- That is, periodic convolution of two sequences multiplies their corresponding Fourier series coefficient sequences (up to a scale factor).

Multiplication

- Let x and y be N-periodic sequences. If $x(n) \overset{\text{DTFS}}{\longleftrightarrow} a_k$ and $y(n) \overset{\text{DTFS}}{\longleftrightarrow} b_k$, then

$$x(n)y(n) \overset{\text{DTFS}}{\longleftrightarrow} a \circledast b(k).$$

- That is, multiplying two sequences results in a circular convolution of their corresponding Fourier series coefficient sequences.

Parseval's Relation

- A sequence x and its Fourier series coefficient sequence a satisfy the following relationship:

$$\frac{1}{N} \sum_{n=\langle N \rangle} |x(n)|^2 = \sum_{k=\langle N \rangle} |a_k|^2 .$$

- The above relationship is simply stating that the amount of energy in a single period of x and the amount of energy in a single period of a are equal up to a scale factor.
- In other words, the transformation between a sequence and its Fourier series coefficient sequence preserves energy (up to a scale factor).

Even/Odd Symmetry

- For an N-periodic sequence x with Fourier-series coefficient sequence a, the following properties hold:

$$x \text{ is even} \Leftrightarrow a \text{ is even}; \quad \text{and}$$
$$x \text{ is odd} \Leftrightarrow a \text{ is odd}.$$

- In other words, the even/odd symmetry properties of x and a always match.

Real Sequences

- A sequence x is ***real*** if and only if its Fourier series coefficient sequence a satisfies

$$a_k = a_{-k}^* \text{ for all } k$$

(i.e., a is ***conjugate symmetric***).

- From properties of complex numbers, one can show that $a_k = a_{-k}^*$ is equivalent to

$$|a_k| = |a_{-k}| \quad \text{and} \quad \arg a_k = -\arg a_{-k}$$

(i.e., $|a_k|$ is ***even*** and $\arg a_k$ is ***odd***).

- Note that x being real does ***not*** necessarily imply that a is real.

Trigonometric Form of a Fourier Series

- Consider the N-periodic sequence x with Fourier series coefficient sequence a.
- If x is real, then its Fourier series can be rewritten in trigonometric form as shown below.
- The **trigonometric form** of a Fourier series has the appearance

$$x(n) = \begin{cases} \alpha_0 + \displaystyle\sum_{k=1}^{N/2-1} \left[\alpha_k \cos\left(\frac{2\pi kn}{N}\right) + \beta_k \sin\left(\frac{2\pi kn}{N}\right)\right] + \\ \quad \alpha_{N/2}\cos(\pi n) & N \text{ even} \\ \alpha_0 + \displaystyle\sum_{k=1}^{(N-1)/2} \left[\alpha_k \cos\left(\frac{2\pi kn}{N}\right) + \beta_k \sin\left(\frac{2\pi kn}{N}\right)\right] & N \text{ odd,} \end{cases}$$

  where $\alpha_0 = a_0$, $\alpha_{N/2} = a_{N/2}$, $\alpha_k = 2\operatorname{Re} a_k$, and $\beta_k = -2\operatorname{Im} a_k$.
- Note that the above trigonometric form contains only ***real*** quantities.

Other Properties of Fourier Series

- For an N-periodic sequence x with Fourier-series coefficient sequence a, the following properties hold:
  1. a_0 is the average value of x over a single period;
  2. x is real and even $\Leftrightarrow$ a is real and even; and
  3. x is real and odd $\Leftrightarrow$ a is purely imaginary and odd.

Section 10.3

Discrete Fourier Transform (DFT)

Prelude to the Discrete Fourier Transform (DFT)

- Letting $a'_k = Na_k$, we can rewrite the Fourier series synthesis and analysis equations, respectively, as

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} a'_k e^{j(2\pi/N)kn} \quad \text{and} \quad a'_k = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)kn}.$$

- Since x and a' are both N-periodic, each of these sequences is completely characterized by its N samples over a single period.
- If we only consider the behavior of x and a' over a single period, this leads to the equations

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} a'_k e^{j(2\pi/N)kn} \quad \text{for } n \in [0\,..\,N-1] \quad \text{and}$$

$$a'_k = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)kn} \quad \text{for } k \in [0\,..\,N-1].$$

- As it turns out, the above two equations define what is known as the discrete Fourier transform (DFT).

Discrete Fourier Transform (DFT)

- The **discrete Fourier transform (DFT)** X of the N-element sequence x is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)kn} \quad \text{for } k \in [0\,..\,N-1].$$

- The preceding equation is known as the **DFT analysis equation**.
- The **inverse DFT** x of the N-element sequence X is given by

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j(2\pi/N)kn} \quad \text{for } n \in [0\,..\,N-1].$$

- The preceding equation is known as the **DFT synthesis equation**.
- The DFT maps a finite-length sequence of N samples to another finite-length sequence of N samples.
- The DFT will be considered in more detail later.

Properties of Discrete Fourier Transform (DFT)

Property	Time Domain	Fourier Domain
Linearity	$a_1 x_1(n) + a_2 x_2(n)$	$a_1 X_1(k) + a_2 X_2(k)$
Translation	$x(n - n_0)$	$e^{-jk(2\pi/N)n_0} X(k)$
Modulation	$e^{j(2\pi/N)k_0 n} x(n)$	$X(k - k_0)$
Reflection	$x(-n)$	$X(-k)$
Conjugation	$x^*(n)$	$X^*(-k)$
Duality	$X(n)$	$Nx(-k)$
Periodic Convolution	$x_1 \circledast x_2(n)$	$X_1(k)X_2(k)$
Multiplication	$x_1(n)x_2(n)$	$\frac{1}{N} X_1 \circledast X_2(k)$

Property	
Parseval's Relation	$\sum_{n=0}^{N-1} \lvert x(n) \rvert^2 = \frac{1}{N} \sum_{k=0}^{N-1} \lvert X(k) \rvert^2$
Even Symmetry	x is even $\Leftrightarrow$ X is even
Odd Symmetry	x is odd $\Leftrightarrow$ X is odd
Real / Conjugate Symmetry	x is real $\Leftrightarrow$ X is conjugate symmetric

Section 10.4

Fourier Series and Frequency Spectra

A New Perspective on Sequences: The Frequency Domain

- The Fourier series provides us with an entirely new way to view sequences.
- Instead of viewing a sequence as having information distributed with respect to ***time*** (i.e., a function whose domain is time), we view a sequence as having information distributed with respect to ***frequency*** (i.e., a function whose domain is frequency).
- This so called frequency-domain perspective is of fundamental importance in engineering.
- Many engineering problems can be solved ***much more easily*** using the frequency domain than the time domain.
- The Fourier series coefficients of a sequence x provide a means to ***quantify*** how much information x has at different frequencies.
- The distribution of information in a sequence over different frequencies is referred to as the ***frequency spectrum*** of the sequence.

Fourier Series and Frequency Spectra

- To gain further insight into the role played by the Fourier series coefficients a_k in the context of the frequency spectrum of the N-periodic sequence x, it is helpful to write the Fourier series with the a_k expressed in ***polar form*** as

$$x(n) = \sum_{k=0}^{N-1} a_k e^{j(2\pi/N)kn} = \sum_{k=0}^{N-1} |a_k| \, e^{j([2\pi/N]kn + \arg a_k)}.$$

- Clearly, the kth term in the summation corresponds to a complex sinusoid with fundamental frequency $\frac{2\pi}{N}k$ that has been ***amplitude scaled*** by a factor of $|a_k|$ and ***time-shifted*** by an amount that depends on $\arg a_k$.
- For a given k, the ***larger*** $|a_k|$ is, the larger is the amplitude of its corresponding complex sinusoid $e^{j(2\pi/N)kn}$, and therefore the ***larger the contribution*** the kth term (which is associated with frequency $\frac{2\pi}{N}k$) will make to the overall summation.
- In this way, we can use $|a_k|$ as a ***measure*** of how much information a sequence x has at the frequency $\frac{2\pi}{N}k$.

Fourier Series and Frequency Spectra (Continued 1)

- The Fourier series coefficients a_k of the sequence x are referred to as the **frequency spectrum** of x.
- The magnitudes $|a_k|$ of the Fourier series coefficients a_k are referred to as the **magnitude spectrum** of x.
- The arguments $\arg a_k$ of the Fourier series coefficients a_k are referred to as the **phase spectrum** of x.
- The frequency spectrum a_k of an N-periodic sequence is N-periodic in the coefficient index k and 2π-periodic in the frequency $\Omega = \frac{2\pi}{N}k$.
- The range of frequencies between $-\pi$ and π are referred to as the **baseband**.
- Often, the spectrum of a sequence is plotted against frequency $\Omega = \frac{2\pi}{N}k$ (over the single 2π period of the baseband) instead of the Fourier series coefficient index k.

Fourier Series and Frequency Spectra (Continued 2)

- Since the Fourier series only has frequency components at integer multiples of the fundamental frequency, the frequency spectrum is ***discrete*** in the independent variable (i.e., frequency).
- Due to the general appearance of frequency-spectrum plot (i.e., a number of vertical lines at various frequencies), we refer to such spectra as **line spectra**.

Frequency Spectra of Real Sequences

- Let x denote an N-periodic sequence with the corresponding Fourier-series coefficient sequence c.
- As we saw earlier:

$$x \text{ is real} \Leftrightarrow c \text{ is conjugate symmetric.}$$

- Furthermore, if x is real, the following assertions hold for c_k for $k \in [0\,..\,N-1]$:
  1. $c_k = c_{N-k}^*$ for $k \in [1\,..\,N-1]$;
  2. of the N coefficients c_k for $k \in [0\,..\,N-1]$, only $\left\lfloor \frac{N}{2} \right\rfloor + 1$ coefficients are independent; for example, c_k for $k \in \left[0\,..\,\left\lfloor \frac{N}{2} \right\rfloor\right]$ completely determines c_k for all $k \in [0\,..\,N-1]$;
  3. c_0 is real; and
  4. if N is even, $c_{N/2}$ is real.
- Note that approximately half of the coefficients in a single period of c are redundant if x is real.

Section 10.5

Fourier Series and LTI Systems

Frequency Response

- Recall that a LTI system $\mathcal{H}$ with impulse response h is such that $\mathcal{H}\{z^n\}(n) = H_Z(z)z^n$, where $H_Z(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$. (That is, complex exponentials are ***eigensequences*** of LTI systems.)
- Since a complex sinusoid is a ***special case*** of a complex exponential, we can reuse the above result for the special case of complex sinusoids.
- For a LTI system $\mathcal{H}$ with impulse response h,

$$\mathcal{H}\left\{e^{j\Omega n}\right\}(n) = H(\Omega)e^{j\Omega n},$$

  where Ω is real and

$$H(\Omega) = \sum_{n=-\infty}^{\infty} h(n)e^{-j\Omega n}.$$

- That is, $e^{j\Omega n}$ is an ***eigensequence*** of a LTI system and $H(\Omega)$ is the corresponding ***eigenvalue***.
- The function H is ***2π-periodic***, since $e^{j\Omega}$ is 2π-periodic.
- We refer to H as the **frequency response** of the system $\mathcal{H}$.

- Consider a LTI system with input x, output y, and frequency response H.
- Suppose that the N-periodic input x is expressed as the Fourier series

$$x(n) = \sum_{k=0}^{N-1} a_k e^{jk\Omega_0 n}, \quad \text{where } \Omega_0 = \frac{2\pi}{N}.$$

- Using our knowledge about the ***eigensequences*** of LTI systems, we can conclude

$$y(n) = \sum_{k=0}^{N-1} a_k H(k\Omega_0) e^{jk\Omega_0 n}.$$

- Thus, if the input x to a LTI system is a Fourier series, the output y is also a Fourier series. More specifically, if $x(n) \overset{\text{DTFS}}{\longleftrightarrow} a_k$ then $y(n) \overset{\text{DTFS}}{\longleftrightarrow} H(k\Omega_0)a_k$.
- The above formula can be used to determine the output of a LTI system from its input in a way that ***does not require convolution***.

Filtering

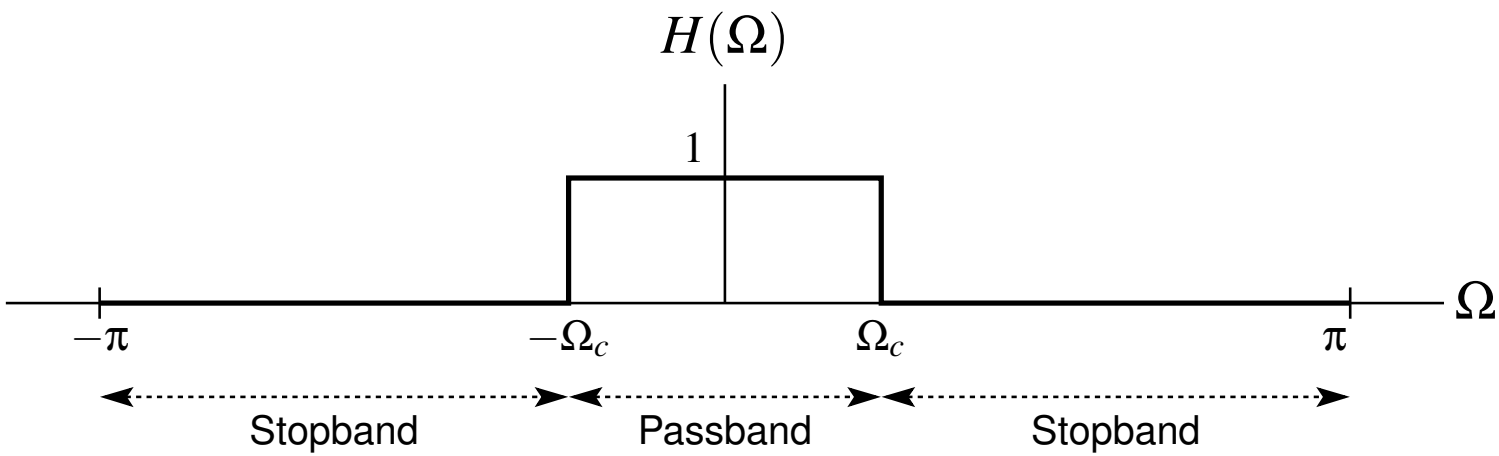
- In many applications, we want to ***modify the spectrum*** of a sequence by either amplifying or attenuating certain frequency components.
- This process of modifying the frequency spectrum of a sequence is called **filtering**.
- A system that performs a filtering operation is called a **filter**.
- Many types of filters exist.
- **Frequency selective filters** pass some frequencies with little or no distortion, while significantly attenuating other frequencies.
- Several basic types of frequency-selective filters include: lowpass, highpass, and bandpass.

Ideal Lowpass Filter

- An **ideal lowpass filter** eliminates all baseband frequency components with a frequency whose magnitude is greater than some cutoff frequency, while leaving the remaining baseband frequency components unaffected.
- Such a filter has a ***frequency response*** of the form

$$H(\Omega) = \begin{cases} 1 & |\Omega| \leq \Omega_c \\ 0 & \Omega_c < |\Omega| \leq \pi, \end{cases}$$

  where Ω_c is the **cutoff frequency**.
- A plot of this frequency response is given below.

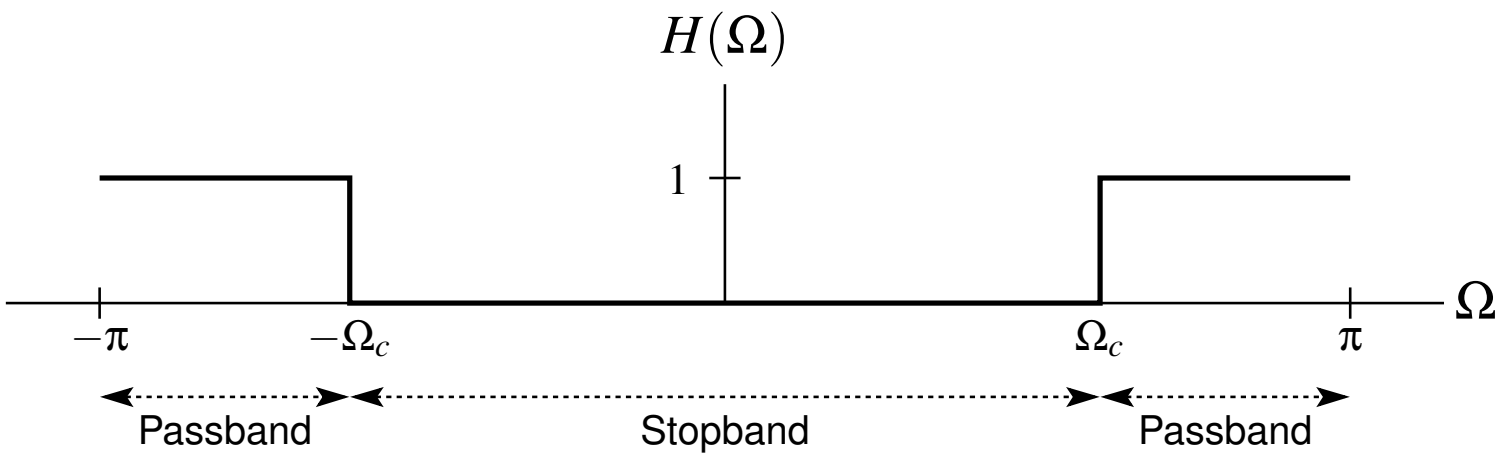


Ideal Highpass Filter

- An **ideal highpass filter** eliminates all baseband frequency components with a frequency whose magnitude is less than some cutoff frequency, while leaving the remaining baseband frequency components unaffected.
- Such a filter has a ***frequency response*** of the form

$$H(\Omega) = \begin{cases} 1 & \Omega_c < |\Omega| \leq \pi \\ 0 & |\Omega| \leq \Omega_c, \end{cases}$$

  where Ω_c is the **cutoff frequency**.
- A plot of this frequency response is given below.

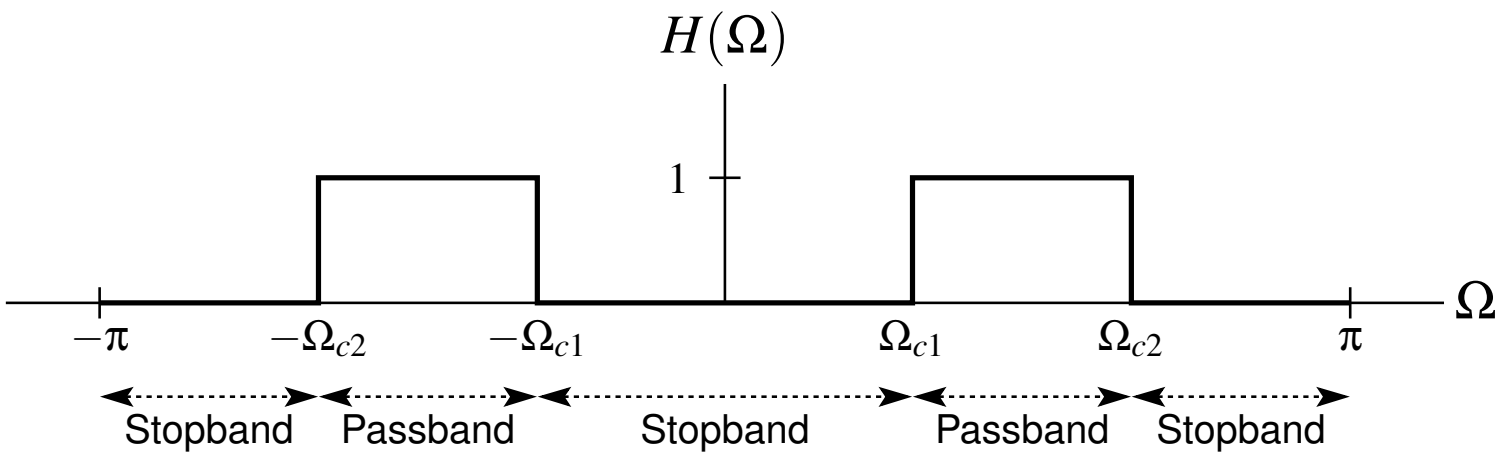


Ideal Bandpass Filter

- An **ideal bandpass filter** eliminates all baseband frequency components with a frequency whose magnitude does not lie in a particular range, while leaving the remaining baseband frequency components unaffected.
- Such a filter has a ***frequency response*** of the form

$$H(\Omega) = \begin{cases} 1 & \Omega_{c1} \leq |\Omega| \leq \Omega_{c2} \\ 0 & |\Omega| < \Omega_{c1} \text{ or } \Omega_{c2} < |\Omega| < \pi, \end{cases}$$

  where the limits of the passband are Ω_{c1} and Ω_{c2}.
- A plot of this frequency response is given below.



Part 11

Discrete-Time Fourier Transform (DTFT)

Motivation for the Fourier Transform

- The (DT) Fourier series provide an extremely useful representation for periodic sequences.
- Often, however, we need to deal with sequences that are not periodic.
- A more general tool than the Fourier series is needed in this case.
- The (DT) Fourier transform can be used to represent both periodic and aperiodic sequences.
- Since the (DT) Fourier transform is essentially derived from (DT) Fourier series through a limiting process, the Fourier transform has many similarities with Fourier series.

Section 11.1

Fourier Transform

Development of the Fourier Transform [Aperiodic Case]

- The (DT) Fourier series is an extremely useful signal representation.
- Unfortunately, this signal representation can only be used for periodic sequences, since a Fourier series is inherently periodic.
- Many sequences are not periodic, however.
- Rather than abandoning Fourier series, one might wonder if we can somehow use Fourier series to develop a representation that can also be applied to aperiodic sequences.
- By viewing an aperiodic sequence as the limiting case of an N-periodic sequence where $N \to \infty$, we can use the Fourier series to develop a signal representation that can be used for aperiodic sequences, known as the Fourier transform.

- Recall that the Fourier series representation of an N-periodic sequence x is given by

$$x(n) = \sum_{k=\langle N \rangle} \underbrace{\left(\frac{1}{N} \sum_{\ell=\langle N \rangle} x(\ell) e^{-j(2\pi/N)k\ell} \right)}_{c_k} e^{j(2\pi/N)kn}.$$

- In the above representation, if we take the limit as $N \to \infty$, we obtain

$$x(n) = \frac{1}{2\pi} \int_{2\pi} \underbrace{\left(\sum_{\ell=-\infty}^{\infty} x(\ell) e^{-j\Omega\ell} \right)}_{X(\Omega)} e^{j\Omega n} d\Omega$$

  (i.e., as $N \to \infty$, the two finite summations become an integral and infinite summation, $\frac{1}{N}$ becomes $\frac{1}{2\pi} d\Omega$, and $\left(\frac{2\pi}{N}\right) k$ becomes Ω).

- This representation for aperiodic sequences is known as the Fourier transform representation.

Generalized Fourier Transform

- The classical Fourier transform for aperiodic sequences does not exist (i.e., $\sum_{n=-\infty}^{\infty} x(n)e^{-j\Omega n}$ fails to converge) for some sequences of great practical interest, such as:
  - a nonzero constant sequence;
  - a periodic sequence (e.g., a real or complex sinusoid); and
  - the unit-step sequence (i.e., u).
- Fortunately, the Fourier transform can be extended to handle such sequences, resulting in what is known as the **generalized Fourier transform**.
- For our purposes, we can think of the classical and generalized Fourier transforms as being defined by the same formulas.
- Therefore, in what follows, we will not typically make a distinction between the classical and generalized Fourier transforms.

DT Fourier Transform (DTFT)

- The **Fourier transform** of the sequence x, denoted $\mathcal{F}x$ or X, is given by

$$\mathcal{F}x(\Omega) = X(\Omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\Omega n}.$$

- The preceding equation is sometimes referred to as **Fourier transform analysis equation** (or **forward Fourier transform equation**).
- The **inverse Fourier transform** of X, denoted $\mathcal{F}^{-1}X$ or x, is given by

$$\mathcal{F}^{-1}X(n) = x(n) = \frac{1}{2\pi}\int_{2\pi} X(\Omega)e^{j\Omega n}d\Omega.$$

- The preceding equation is sometimes referred to as the **Fourier transform synthesis equation** (or **inverse Fourier transform equation**).
- As a matter of notation, to denote that a sequence x has the Fourier transform X, we write $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$.
- A sequence x and its Fourier transform X constitute what is called a **Fourier transform pair**.

Section 11.2

Convergence Properties of the Fourier Transform

Convergence of the Fourier Transform

- For a sequence x, the Fourier transform analysis equation (i.e., $X(\Omega) = \sum_{-\infty}^{\infty} x(n) e^{-j\Omega n}$) converges ***uniformly*** if

$$\sum_{k=-\infty}^{\infty} |x(k)| < \infty$$

  (i.e., x is ***absolutely summable***).

- For a sequence x, the Fourier transform analysis equation (i.e., $X(\Omega) = \sum_{-\infty}^{\infty} x(n) e^{-j\Omega n}$) converges in the ***MSE sense*** if

$$\sum_{k=-\infty}^{\infty} |x(k)|^2 < \infty$$

  (i.e., x is ***square summable***).

- For a bounded Fourier transform X, the Fourier transform synthesis equation (i.e., $x(n) = \frac{1}{2\pi} \int_{2\pi} X(\Omega) e^{j\Omega n} d\Omega$) will always converge, since the integration interval is finite.

Section 11.3

Properties of the Fourier Transform

Properties of the (DT) Fourier Transform

Property	Time Domain	Frequency Domain
Linearity	$a_1 x_1(n) + a_2 x_2(n)$	$a_1 X_1(\Omega) + a_2 X_2(\Omega)$
Translation	$x(n - n_0)$	$e^{-j\Omega n_0} X(\Omega)$
Modulation	$e^{j\Omega_0 n} x(n)$	$X(\Omega - \Omega_0)$
Conjugation	$x^*(n)$	$X^*(-\Omega)$
Time Reversal	$x(-n)$	$X(-\Omega)$
Upsampling	$(\uparrow M)x(n)$	$X(M\Omega)$
Downsampling	$(\downarrow M)x(n)$	$\frac{1}{M}\sum_{k=0}^{M-1} X\left(\frac{\Omega - 2\pi k}{M}\right)$
Convolution	$x_1 * x_2(n)$	$X_1(\Omega)X_2(\Omega)$
Multiplication	$x_1(n)x_2(n)$	$\frac{1}{2\pi}\int_{2\pi} X_1(\theta)X_2(\Omega - \theta)d\theta$
Freq.-Domain Diff.	$nx(n)$	$j\frac{d}{d\Omega}X(\Omega)$
Differencing	$x(n) - x(n-1)$	$\left(1 - e^{-j\Omega}\right)X(\Omega)$
Accumulation	$\sum_{k=-\infty}^{n} x(k)$	$\frac{e^{j\Omega}}{e^{j\Omega}-1}X(\Omega) + \pi X(0)\sum_{k=-\infty}^{\infty}\delta(\Omega - 2\pi k)$

Properties of the (DT) Fourier Transform (Continued)

Property	
Periodicity	$X(\Omega) = X(\Omega + 2\pi)$
Parseval's Relation	$\sum_{n=-\infty}^{\infty} \lvert x(n) \rvert^2 = \frac{1}{2\pi} \int_{2\pi} \lvert X(\Omega) \rvert^2 \, d\Omega$
Even Symmetry	x is even $\Leftrightarrow$ X is even
Odd Symmetry	x is odd $\Leftrightarrow$ X is odd
Real / Conjugate Symmetry	x is real $\Leftrightarrow$ X is conjugate symmetric

(DT) Fourier Transform Pairs

Pair	$x(n)$	$X(\Omega)$
1	$\delta(n)$	1
2	1	$2\pi\sum_{k=-\infty}^{\infty}\delta(\Omega-2\pi k)$
3	$u(n)$	$\frac{e^{j\Omega}}{e^{j\Omega}-1}+\sum_{k=-\infty}^{\infty}\pi\delta(\Omega-2\pi k)$
4	$a^n u(n), \lvert a\rvert<1$	$\frac{e^{j\Omega}}{e^{j\Omega}-a}$
5	$-a^n u(-n-1), \lvert a\rvert>1$	$\frac{e^{j\Omega}}{e^{j\Omega}-a}$
6	$a^{\lvert n\rvert}, \lvert a\rvert<1$	$\frac{1-a^2}{1-2a\cos\Omega+a^2}$
7	$\cos(\Omega_0 n)$	$\pi\sum_{k=-\infty}^{\infty}[\delta(\Omega-\Omega_0-2\pi k)+\delta(\Omega+\Omega_0-2\pi k)]$
8	$\sin(\Omega_0 n)$	$j\pi\sum_{k=-\infty}^{\infty}[\delta(\Omega+\Omega_0-2\pi k)-\delta(\Omega-\Omega_0-2\pi k)]$
9	$\cos(\Omega_0 n)u(n)$	$\frac{e^{j2\Omega}-e^{j\Omega}\cos\Omega_0}{e^{j2\Omega}-2e^{j\Omega}\cos\Omega_0+1}+\frac{\pi}{2}\sum_{k=-\infty}^{\infty}[\delta(\Omega-2\pi k-\Omega_0)+\delta(\Omega-2\pi k+\Omega_0)]$
10	$\sin(\Omega_0 n)u(n)$	$\frac{e^{j\Omega}\sin\Omega_0}{e^{j2\Omega}-2e^{j\Omega}\cos\Omega_0+1}+\frac{\pi}{2j}\sum_{k=-\infty}^{\infty}[\delta(\Omega-2\pi k-\Omega_0)-\delta(\Omega-2\pi k+\Omega_0)]$
11	$\frac{B}{\pi}\operatorname{sinc}(Bn), 0<B<\pi$	$\sum_{k=-\infty}^{\infty}\operatorname{rect}\left(\frac{\Omega-2\pi k}{2B}\right)$
12	$u(n)-u(n-M)$	$e^{-j\Omega(M-1)/2}\left(\frac{\sin(M\Omega/2)}{\sin(\Omega/2)}\right)$
13	$na^n u(n), \lvert a\rvert<1$	$\frac{ae^{j\Omega}}{(e^{j\Omega}-a)^2}$

Periodicity

- Recall the definition of the Fourier transform X of the sequence x:

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\Omega n}.$$

- For all integer k, we have that

$$\begin{aligned} X(\Omega + 2\pi k) &= \sum_{n=-\infty}^{\infty} x(n)e^{-j(\Omega+2\pi k)n} \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j(\Omega n+2\pi kn)} \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j\Omega n} \\ &= X(\Omega). \end{aligned}$$

- Thus, the Fourier transform X of the sequence x is always ***2π-periodic***.

Linearity

- If $x_1(n) \overset{\text{DTFT}}{\longleftrightarrow} X_1(\Omega)$ and $x_2(n) \overset{\text{DTFT}}{\longleftrightarrow} X_2(\Omega)$, then

$$a_1 x_1(n) + a_2 x_2(n) \overset{\text{DTFT}}{\longleftrightarrow} a_1 X_1(\Omega) + a_2 X_2(\Omega),$$

  where a_1 and a_2 are arbitrary complex constants.
- This is known as the **linearity property** of the Fourier transform.

Translation

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$x(n-n_0) \overset{\text{DTFT}}{\longleftrightarrow} e^{-j\Omega n_0} X(\Omega),$$

  where n_0 is an arbitrary integer.
- This is known as the **translation (or time-domain shifting) property** of the Fourier transform.

Modulation

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$e^{j\Omega_0 n} x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega - \Omega_0),$$

  where Ω_0 is an arbitrary real constant.
- This is known as the **modulation (or frequency-domain shifting) property** of the Fourier transform.

Conjugation

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$x^*(n) \overset{\text{DTFT}}{\longleftrightarrow} X^*(-\Omega).$$

- This is known as the **conjugation property** of the Fourier transform.

Time Reversal

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$x(-n) \overset{\text{DTFT}}{\longleftrightarrow} X(-\Omega).$$

- This is known as the **time-reversal property** of the Fourier transform.

Upsampling

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$(\uparrow M)x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(M\Omega).$$

- This is known as the **upsampling property** of the Fourier transform.

Downsampling

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$(\downarrow M)x(n) \overset{\text{DTFT}}{\longleftrightarrow} \frac{1}{M} \sum_{k=0}^{M-1} X\left(\frac{\Omega - 2\pi k}{M}\right).$$

- This is known as the **downsampling property** of the Fourier transform.

Convolution

- If $x_1(n) \overset{\text{DTFT}}{\longleftrightarrow} X_1(\Omega)$ and $x_2(n) \overset{\text{DTFT}}{\longleftrightarrow} X_2(\Omega)$, then

$$x_1 * x_2(n) \overset{\text{DTFT}}{\longleftrightarrow} X_1(\Omega)X_2(\Omega).$$

- This is known as the **convolution (or time-domain convolution) property** of the Fourier transform.
- In other words, a convolution in the time domain becomes a multiplication in the frequency domain.
- This suggests that the Fourier transform can be used to avoid having to deal with convolution operations.

Multiplication

- If $x_1(n) \overset{\text{DTFT}}{\longleftrightarrow} X_1(\Omega)$ and $x_2(n) \overset{\text{DTFT}}{\longleftrightarrow} X_2(\Omega)$, then

$$x_1(n)x_2(n) \overset{\text{DTFT}}{\longleftrightarrow} \frac{1}{2\pi}\int_{2\pi} X_1(\theta)X_2(\Omega-\theta)d\theta.$$

- This is known as the **multiplication (or time-domain multiplication) property** of the Fourier transform.
- Do not forget the factor of $\frac{1}{2\pi}$ in the above formula!
- This property of the Fourier transform is often tedious to apply (in the forward direction) as it turns a multiplication into a convolution.

Frequency-Domain Differentiation

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$nx(n) \overset{\text{DTFT}}{\longleftrightarrow} j\frac{d}{d\Omega}X(\Omega).$$

- This is known as the **frequency-domain differentiation property** of the Fourier transform.

Differencing

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$x(n) - x(n-1) \overset{\text{DTFT}}{\longleftrightarrow} \left(1 - e^{-j\Omega}\right) X(\Omega).$$

- This is known as the **differencing property** of the Fourier transform.
- Note that this property follows quite trivially from the linearity and translation properties of the Fourier transform.

Accumulation

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$\sum_{k=-\infty}^{n} x(k) \overset{\text{DTFT}}{\longleftrightarrow} \frac{e^{j\Omega}}{e^{j\Omega}-1} X(\Omega) + \pi X(0) \sum_{k=-\infty}^{\infty} \delta(\Omega - 2\pi k).$$

- This is known as the **accumulation property** of the Fourier transform.

Parseval's Relation

- If $x(n) \overset{\text{DTFT}}{\longleftrightarrow} X(\Omega)$, then

$$\sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi} \int_{2\pi} |X(\Omega)|^2 \, d\Omega$$

  (i.e., the energy of x and energy of X are equal up to a factor of 2π).
- This is known as **Parseval's relation**.
- Since energy is often a quantity of great significance in engineering applications, it is extremely helpful to know that the Fourier transform *preserves energy* (up to a scale factor).

Even and Odd Symmetry

- For a sequence x with Fourier transform X, the following assertions hold:
  1. x is even $\Leftrightarrow$ X is even; and
  2. x is odd $\Leftrightarrow$ X is odd.
- In other words, the forward and inverse Fourier transforms preserve even/odd symmetry.

Real Sequences

- A sequence x is ***real*** if and only if its Fourier transform X satisfies

$$X(\Omega) = X^*(-\Omega) \text{ for all } \Omega$$

  (i.e., X is ***conjugate symmetric***).
- Thus, for a real-valued sequence, the portion of the graph of a Fourier transform for negative values of frequency Ω is ***redundant***, as it is completely determined by symmetry.
- From properties of complex numbers, one can show that $X(\Omega) = X^*(-\Omega)$ is equivalent to

$$|X(\Omega)| = |X(-\Omega)| \quad \text{and} \quad \arg X(\Omega) = -\arg X(-\Omega)$$

  (i.e., $|X(\Omega)|$ is ***even*** and $\arg X(\Omega)$ is ***odd***).
- Note that x being real does ***not*** necessarily imply that X is real.

Section 11.4

Fourier Transform of Periodic Sequences

Fourier Transform of Periodic Sequences

- The Fourier transform can be generalized to also handle periodic sequences.
- Consider an N-periodic sequence x.
- Define the sequence x_N as

$$x_N(n) = \begin{cases} x(n) & 0 \leq n < N \\ 0 & \text{otherwise.} \end{cases}$$

  (i.e., $x_N(n)$ is equal to $x(n)$ over a single period and zero elsewhere).
- Let a denote the Fourier series coefficient sequence of x.
- Let X and X_N denote the Fourier transforms of x and x_N, respectively.
- The following relationships can be shown to hold:

$$X(\Omega) = \frac{2\pi}{N} \sum_{k=-\infty}^{\infty} X_N\left(\frac{2\pi k}{N}\right) \delta\left(\Omega - \frac{2\pi k}{N}\right),$$

$$a_k = \frac{1}{N} X_N\left(\frac{2\pi k}{N}\right), \quad \text{and} \quad X(\Omega) = 2\pi \sum_{k=-\infty}^{\infty} a_k \delta\left(\Omega - \frac{2\pi k}{N}\right).$$

Fourier Transform of Periodic Sequences (Continued)

- The Fourier series coefficient sequence a is produced by sampling X_N at integer multiples of the fundamental frequency $\frac{2\pi}{N}$ and scaling the resulting sequence by $\frac{1}{N}$.
- The Fourier transform of a periodic sequence can only be nonzero at integer multiples of the fundamental frequency.

Section 11.5

Fourier Transform and Frequency Spectra of Sequences

Frequency Spectra of Sequences

- Like Fourier series, the Fourier transform also provides us with a frequency-domain perspective on sequences.
- That is, instead of viewing a sequence as having information distributed with respect to ***time*** (i.e., a function whose domain is time), we view a sequence as having information distributed with respect to ***frequency*** (i.e., a function whose domain is frequency).
- The Fourier transform X of a sequence x provides a means to ***quantify*** how much information x has at different frequencies.
- The distribution of information in a sequence over different frequencies is referred to as the ***frequency spectrum*** of the sequence.

- To gain further insight into the role played by the Fourier transform X in the context of the frequency spectrum of x, it is helpful to write the Fourier transform representation of x with $X(\Omega)$ expressed in ***polar form*** as follows:

$$x(n) = \frac{1}{2\pi}\int_{2\pi} X(\Omega)e^{j\Omega n}d\Omega = \frac{1}{2\pi}\int_{2\pi} |X(\Omega)|\, e^{j[\Omega n + \arg X(\Omega)]}d\Omega.$$

- In effect, the quantity $|X(\Omega)|$ is a ***weight*** that determines how much the complex sinusoid at frequency Ω contributes to the integration result $x(n)$.
- Perhaps, this can be more easily seen if we express the above integral as the ***limit of a sum***, derived from an approximation of the integral using the area of rectangles, as shown on the next slide. [Recall that $\int_a^b f(x)dx = \lim_{n\to\infty}\sum_{k=1}^{n} f(x_k)\Delta x$ where $\Delta x = \frac{b-a}{n}$ and $x_k = a + k\Delta x$.]

Fourier Transform and Frequency Spectra (Continued 1)

- Expressing the integral (from the previous slide) as the ***limit of a sum***, we obtain

$$x(n) = \lim_{\ell \to \infty} \frac{1}{2\pi} \sum_{k=1}^{\ell} \Delta\Omega \left| X(\Omega') \right| e^{j[\Omega' n + \arg X(\Omega')]},$$

  where $\Delta\Omega = \frac{2\pi}{\ell}$ and $\Omega' = k\Delta\Omega$.

- In the above equation, the kth term in the summation corresponds to a complex sinusoid with fundamental frequency $\Omega' = k\Delta\Omega$ that has had its ***amplitude scaled*** by a factor of $\left|X(\Omega')\right|$ and has been ***time shifted*** by an amount that depends on $\arg X(\Omega')$.
- For a given $\Omega' = k\Delta\Omega$ (which is associated with the kth term in the summation), the larger $\left|X(\Omega')\right|$ is, the larger the amplitude of its corresponding complex sinusoid $e^{j\Omega' n}$ will be, and therefore the larger the contribution the kth term will make to the overall summation.
- In this way, we can use $\left|X(\Omega')\right|$ as a ***measure*** of how much information a sequence x has at the frequency Ω'.

Fourier Transform and Frequency Spectra (Continued 2)

- The Fourier transform X of the sequence x is referred to as the **frequency spectrum** of x.
- The magnitude $|X(\Omega)|$ of the Fourier transform X is referred to as the **magnitude spectrum** of x.
- The argument $\arg X(\Omega)$ of the Fourier transform X is referred to as the **phase spectrum** of x.
- Since the Fourier transform is a function of a real variable, a sequence can potentially have information at any real frequency.
- Earlier, we saw that for periodic sequences, the Fourier transform can only be nonzero at integer multiples of the fundamental frequency.
- So, the Fourier transform and Fourier series give a consistent picture in terms of frequency spectra.
- Since the frequency spectrum is complex (in the general case), it is ***usually represented using two plots***, one showing the magnitude spectrum and one showing the phase spectrum.

Frequency Spectra of Real Sequences

- Recall that, for a ***real*** sequence x, the Fourier transform X of x satisfies

$$X(\Omega) = X^*(-\Omega)$$

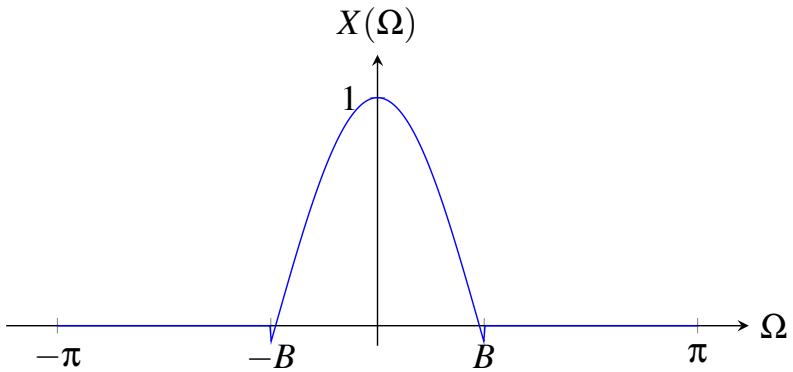
(i.e., X is ***conjugate symmetric***), which is equivalent to

$$|X(\Omega)| = |X(-\Omega)| \quad \text{and} \quad \arg X(\Omega) = -\arg X(-\Omega).$$

- Since $|X(\Omega)| = |X(-\Omega)|$, the magnitude spectrum of a real sequence is always ***even***.
- Similarly, since $\arg X(\Omega) = -\arg X(-\Omega)$, the phase spectrum of a real sequence is always ***odd***.
- Due to the symmetry in the frequency spectra of real sequences, we typically ***ignore negative frequencies*** when dealing with such sequences.
- In the case of sequences that are complex but not real, frequency spectra do not possess the above symmetry, and ***negative frequencies become important***.

Bandwidth

- A sequence x with Fourier transform X satisfying $X(\Omega) = 0$ for all Ω in $(-\pi, \pi]$ except for some interval I is said to be **bandlimited** to frequencies in I.
- The **bandwidth** of a sequence x with Fourier transform X is the length of the interval in $(-\pi, \pi]$ over which X is nonzero.
- For example, the sequence x with the Fourier transform X shown below is bandlimited to frequencies in $[-B, B]$ and has bandwidth $B - (-B) = 2B$.



- Since x is real in the above example (as X is conjugate symmetric), we might choose to ignore negative frequencies, in which case x would be deemed to be bandlimited to frequencies in $[0, B]$ and have bandwidth $B - 0 = B$.

Energy-Density Spectra

- By Parseval's relation, the energy E in a sequence x with Fourier transform X is given by

$$E = \frac{1}{2\pi} \int_{2\pi} E_x(\Omega) d\Omega,$$

where

$$E_x(\Omega) = |X(\Omega)|^2.$$

- We refer to E_x as the **energy-density spectrum** of the sequence x.
- The function E_x indicates how the energy in x is distributed with respect to frequency.
- For example, the energy contributed by frequencies in the range $[\Omega_1, \Omega_2]$ is given by

$$\frac{1}{2\pi} \int_{\Omega_1}^{\Omega_2} E_x(\Omega) d\Omega.$$

Section 11.6

Fourier Transform and LTI Systems

Frequency Response of LTI Systems

- Consider a LTI system with input x, output y, and impulse response h, and let X, Y, and H denote the Fourier transforms of x, y, and h, respectively.
- Since $y(n) = x * h(n)$, we have that

$$Y(\Omega) = X(\Omega)H(\Omega).$$

- The function H is called the **frequency response** of the system.
- A LTI system is ***completely characterized*** by its frequency response H.
- The above equation provides an alternative way of viewing the behavior of a LTI system. That is, we can view the system as operating in the frequency domain on the Fourier transforms of the input and output signals.
- The frequency spectrum of the output is the product of the frequency spectrum of the input and the frequency response of the system.

Frequency Response of LTI Systems (Continued 1)

- In the general case, the frequency response H is a complex-valued function.
- Often, we represent $H(\Omega)$ in terms of its magnitude $|H(\Omega)|$ and argument $\arg H(\Omega)$.
- The quantity $|H(\Omega)|$ is called the **magnitude response** of the system.
- The quantity $\arg H(\Omega)$ is called the **phase response** of the system.
- Since $Y(\Omega) = X(\Omega)H(\Omega)$, we trivially have that

$$|Y(\Omega)| = |X(\Omega)|\,|H(\Omega)| \quad \text{and} \quad \arg Y(\Omega) = \arg X(\Omega) + \arg H(\Omega).$$

- The magnitude spectrum of the output equals the magnitude spectrum of the input times the magnitude response of the system.
- The phase spectrum of the output equals the phase spectrum of the input plus the phase response of the system.

Frequency Response of LTI Systems (Continued 2)

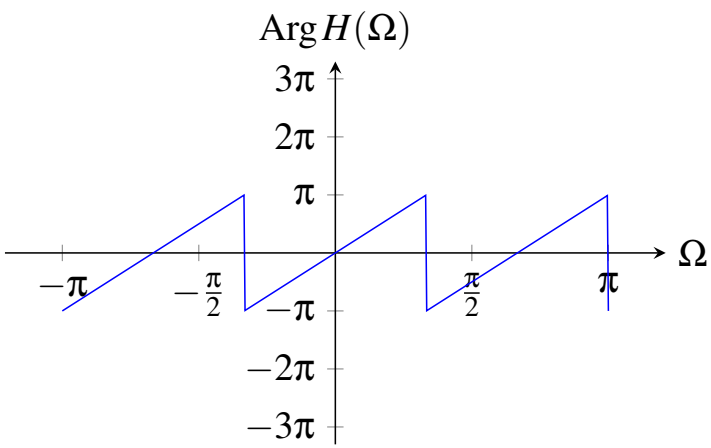
- Since the frequency response H is simply the frequency spectrum of the impulse response h, if h is ***real***, then

$$|H(\Omega)| = |H(-\Omega)| \quad \text{and} \quad \arg H(\Omega) = -\arg H(-\Omega)$$

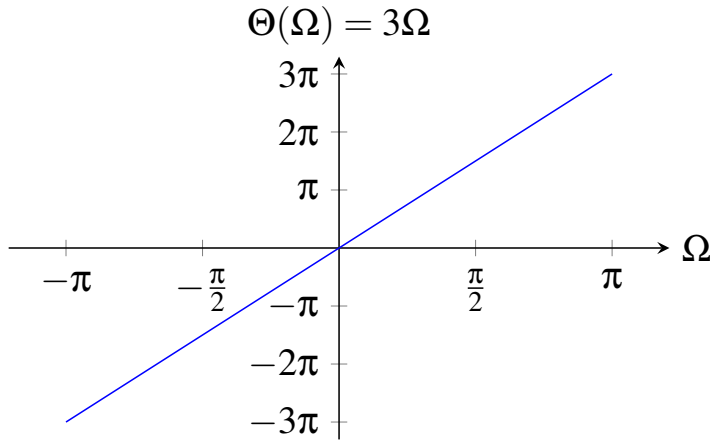
(i.e., the magnitude response $|H(\Omega)|$ is ***even*** and the phase response $\arg H(\Omega)$ is ***odd***).

Unwrapped Phase

- For many types of analysis, restricting the range of a phase function to an interval of length 2π (such as $(-\pi, \pi]$), often unnecessarily introduces discontinuities into the function.
- This motivates the notion of unwrapped phase.
- The **unwrapped phase** is simply the phase defined in such a way so as not to restrict the phase to an interval of length 2π and to keep the phase function continuous to the greatest extent possible.
- For example, the function $H(\Omega) = e^{j3\Omega}$ has the unwrapped phase $\Theta(\Omega) = 3\Omega$.



Phase



Unwrapped Phase

Interpretation of Magnitude and Phase Response

- Recall that a LTI system $\mathcal{H}$ with frequency response H is such that
$$\mathcal{H}\left\{e^{j\Omega n}\right\}(n) = H(\Omega)e^{j\Omega n}.$$
- Expressing $H(\Omega)$ in polar form, we have
$$\begin{aligned}\mathcal{H}\{e^{j\Omega n}\}(n) &= |H(\Omega)|\, e^{j\arg H(\Omega)} e^{j\Omega n} \\ &= |H(\Omega)|\, e^{j[\Omega n + \arg H(\Omega)]} \\ &= |H(\Omega)|\, e^{j\Omega(n+\arg[H(\Omega)]/\Omega)}.\end{aligned}$$
- Thus, the response of the system to the sequence $e^{j\Omega n}$ is produced by applying two transformations to this sequence:
  - (amplitude) scaling by $|H(\Omega)|$; and
  - translating by $-\frac{\arg H(\Omega)}{\Omega}$ (using bandlimited interpolation if $-\frac{\arg H(\Omega)}{\Omega} \notin \mathbb{Z}$).
- Therefore, the magnitude response determines how different complex sinusoids are ***scaled*** (in amplitude) by the system.
- Similarly, the phase response determines how different complex sinusoids are ***translated*** (i.e., delayed/advanced) by the system.

Magnitude Distortion

- Recall that a LTI system $\mathcal{H}$ with frequency response H is such that

$$\mathcal{H}\{e^{j\Omega n}\}(n) = |H(\Omega)|\, e^{j\Omega(n+\arg[H(\Omega)]/\Omega)}.$$

- If $|H(\Omega)|$ is a constant (for all Ω), every complex sinusoid is scaled by the same amount when passing through the system.
- A system for which $|H(\Omega)| = 1$ (for all Ω) is said to be **allpass**.
- In the case of an allpass system, the magnitude spectra of the system's input and output are identical.
- If $|H(\Omega)|$ is not a constant, different complex sinusoids are scaled by different amounts, resulting in what is known as **magnitude distortion**.

Phase Distortion

- Recall that a LTI system $\mathcal{H}$ with frequency response H is such that

$$\mathcal{H}\{e^{j\Omega n}\}(n) = |H(\Omega)|\, e^{j\Omega(n+\arg[H(\Omega)]/\Omega)}.$$

- The preceding equation can be rewritten as

$$\mathcal{H}\{e^{j\Omega n}\}(n) = |H(\Omega)|\, e^{j\Omega[n-\tau_p(\Omega)]} \quad \text{where} \quad \tau_p(\Omega) = -\frac{\arg H(\Omega)}{\Omega}.$$

- The function τ_p is known as the **phase delay** of the system.
- If $\tau_p(\Omega) = n_d$ (where n_d is a constant), the system shifts all complex sinusoids by the same amount n_d.
- Since $\tau_p(\Omega) = n_d$ is equivalent to the (unwrapped) phase response being of the form $\arg H(\Omega) = -n_d\Omega$ (which is a linear function with a zero constant term), a system with a constant phase delay is said to have **linear phase**.
- In the case that $\tau_p(\Omega) = 0$, the system is said to have **zero phase**.
- If $\tau_p(\Omega)$ is not a constant, different complex sinusoids are shifted by different amounts, resulting in what is known as **phase distortion**.

Distortionless Transmission

- Consider a LTI system $\mathcal{H}$ with input x and output y given by

$$y(n) = x(n - n_0),$$

  where n_0 is an integer constant.
- That is, the output of the system is simply the input delayed by n_0.
- This type of behavior is the ideal for which we strive in real-world communication systems (i.e., the received signal y equals a delayed version of the transmitted signal x).
- Taking the Fourier transform of the preceding equation, we have

$$Y(\Omega) = e^{-j\Omega n_0} X(\Omega).$$

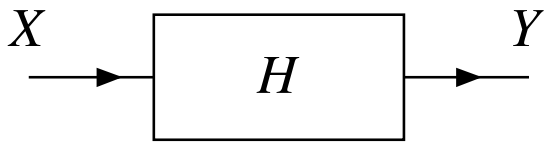
- Thus, the system has the frequency response H given by

$$H(\Omega) = e^{-j\Omega n_0}.$$

- Since the phase delay of the system is $\tau_p(\Omega) = -\left(\frac{-\Omega n_0}{\Omega}\right) = n_0$, the phase delay is constant and the system has linear phase.

Block Diagram Representations of LTI Systems

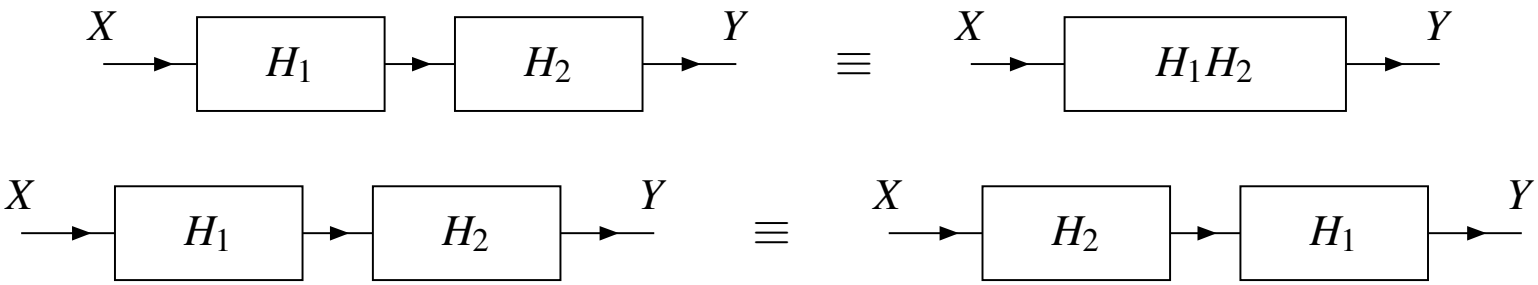
- Consider a LTI system with input x, output y, and impulse response h, and let X, Y, and H denote the Fourier transforms of x, y, and h, respectively.
- Often, it is convenient to represent such a system in block diagram form in the frequency domain as shown below.



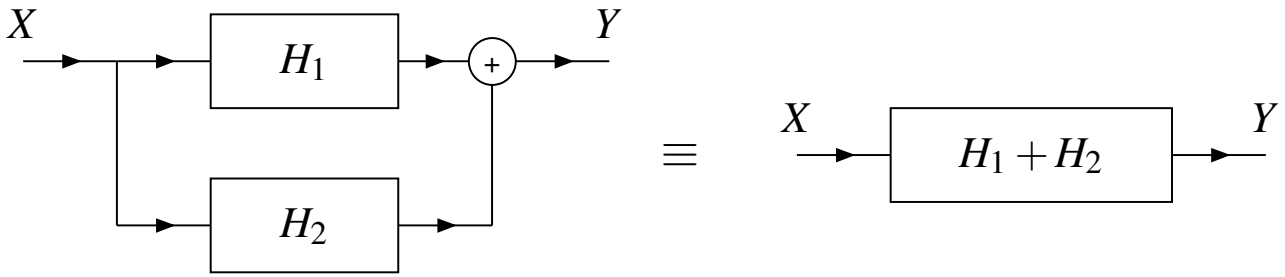
- Since a LTI system is completely characterized by its frequency response, we typically label the system with this quantity.

Interconnection of LTI Systems

- The ***series*** interconnection of the LTI systems with frequency responses H_1 and H_2 is the LTI system with frequency response $H_1 H_2$. That is, we have the equivalences shown below.



- The ***parallel*** interconnection of the LTI systems with frequency responses H_1 and H_2 is the LTI system with the frequency response $H_1 + H_2$. That is, we have the equivalence shown below.



LTI Systems and Difference Equations

- Many LTI systems of practical interest can be represented using an ***Nth-order linear difference equation with constant coefficients***.
- Consider a system with input x and output y that is characterized by an equation of the form

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

- Let h denote the impulse response of the system, and let X, Y, and H denote the Fourier transforms of x, y, and h, respectively.
- One can show that $H(\Omega)$ is given by

$$H(\Omega) = \frac{Y(\Omega)}{X(\Omega)} = \frac{\sum_{k=0}^{M} a_k (e^{j\Omega})^{-k}}{\sum_{k=0}^{N} b_k (e^{j\Omega})^{-k}} = \frac{\sum_{k=0}^{M} a_k e^{-jk\Omega}}{\sum_{k=0}^{N} b_k e^{-jk\Omega}}.$$

- Each of the numerator and denominator of H is a ***polynomial*** in $e^{-j\Omega}$.
- Thus, H is a ***rational function*** in the variable $e^{-j\Omega}$.

Section 11.7

Fourier Transform Relationships

Duality Between DTFT and CTFS

- The DTFT analysis and synthesis equations are, respectively, given by

$$X(\Omega) = \sum_{k=-\infty}^{\infty} x(k)e^{-jk\Omega} \quad \text{and} \quad x(n) = \frac{1}{2\pi}\int_{2\pi} X(\Omega)e^{jn\Omega}d\Omega.$$

- The CTFS synthesis and analysis equations are, respectively, given by

$$x_c(t) = \sum_{k=-\infty}^{\infty} a(k)e^{jk(2\pi/T)t} \quad \text{and} \quad a(n) = \frac{1}{T}\int_{T} x_c(t)e^{-jn(2\pi/T)t}dt,$$

  which can be rewritten, respectively, as

$$x_c(t) = \sum_{k=-\infty}^{\infty} a(-k)e^{-jk(2\pi/T)t} \quad \text{and} \quad a(-n) = \frac{1}{T}\int_{T} x_c(t)e^{jn(2\pi/T)t}dt.$$

- The CTFS synthesis equation with $T = 2\pi$ corresponds to the DTFT analysis equation with $X = x_c$, $\Omega = t$, and $x(n) = a(-n)$.
- The CTFS analysis equation with $T = 2\pi$ corresponds to the DTFT synthesis equation with $X = x_c$ and $x(n) = a(-n)$.
- Consequently, the DTFT X of the sequence x can be viewed as a CTFS representation of the 2π-periodic spectrum X.

Relationship Between DTFT and CTFT

- Let x be a bandlimited function and let T denote a sampling period for x that satisfies the Nyquist condition.
- Let $\tilde{y}$ be the function obtained by impulse sampling x with sampling period T. That is,

$$\tilde{y}(t) = \sum_{n=-\infty}^{\infty} x(Tn)\delta(t - Tn).$$

- Let y denote the sequence obtaining by sampling x with sampling period T. That is,

$$y(n) = x(Tn).$$

- Let $\tilde{Y}$ denote the (CT) Fourier transform of $\tilde{y}$ and let Y denote the (DT) Fourier transform of y.
- Then, the following relationship holds:

$$Y(\Omega) = \tilde{Y}\left(\frac{\Omega}{T}\right) \quad \text{for all } \Omega \in \mathbb{R}.$$

Relationship Between DTFT and DFT

- Let x be a sequence with (DT) Fourier transform X such that

$$x(n) = 0 \quad \text{for all } n \notin [0\,..\,M-1].$$

- Let $\tilde{X}$ denote the N-point DFT of X. That is,

$$\tilde{X}(k) = \sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)kn} \quad \text{for } k \in [0\,..\,N-1].$$

- Suppose now that $N \geq M$.
- Then, the following relationship holds:

$$X\left(\tfrac{2\pi}{N}k\right) = \tilde{X}(k) \quad \text{for } k \in [0\,..\,N-1].$$

- In other words, the elements of the sequence $\tilde{X}$ correspond to uniformly-spaced samples of the function X.

Spectral Sampling Example

- Consider the sequence

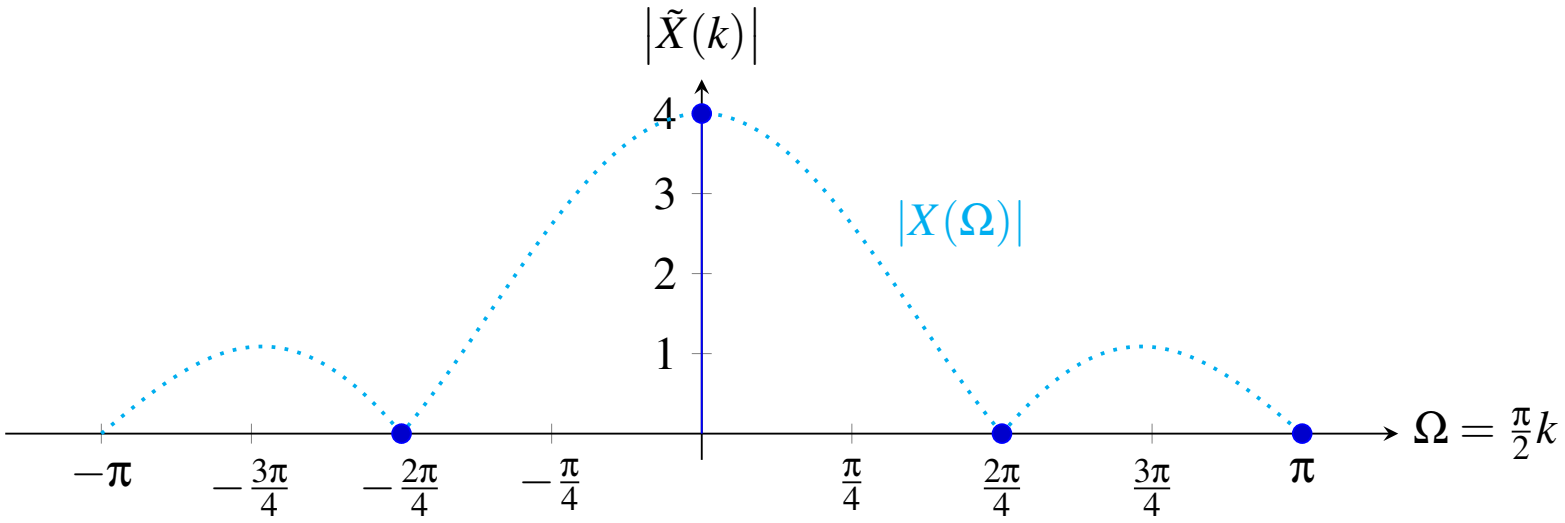
$$x(n) = u(n) - u(n-4).$$

- The Fourier transform X of x can be shown to be

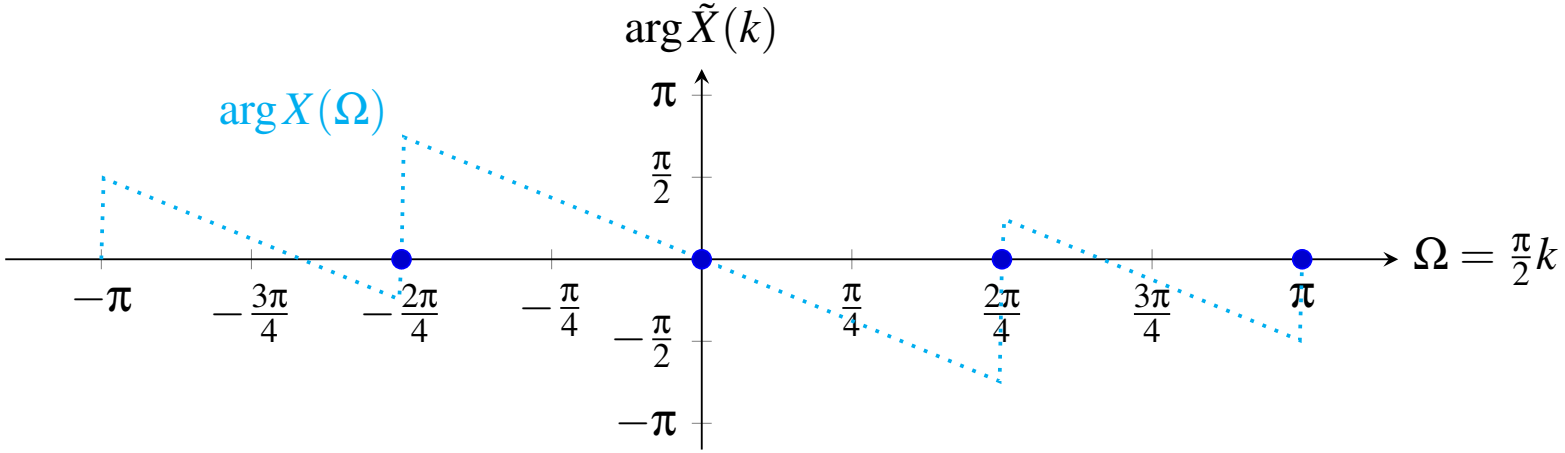
$$X(\Omega) = e^{-j(3/2)\Omega} \left[\frac{\sin(2\Omega)}{\sin\left(\frac{1}{2}\Omega\right)} \right].$$

- Clearly, $x(n) = 0$ for all $n \notin [0\,..\,3]$.
- Therefore, uniformly-spaced samples of X can be obtained from an N-point DFT $\tilde{X}$ of x, where $N \geq 4$.
- The subsequent slides show the sampled spectrum obtained by the DFT for several values of N.

Spectral Sampling Example: $N = 4$

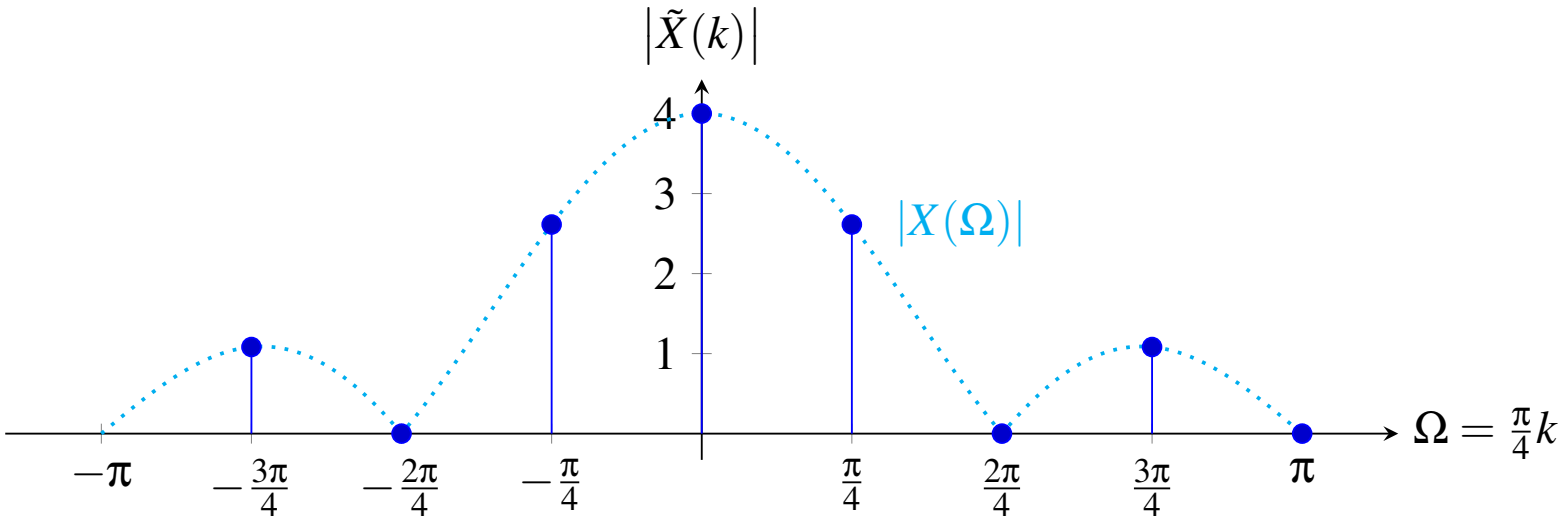


Magnitude Spectrum

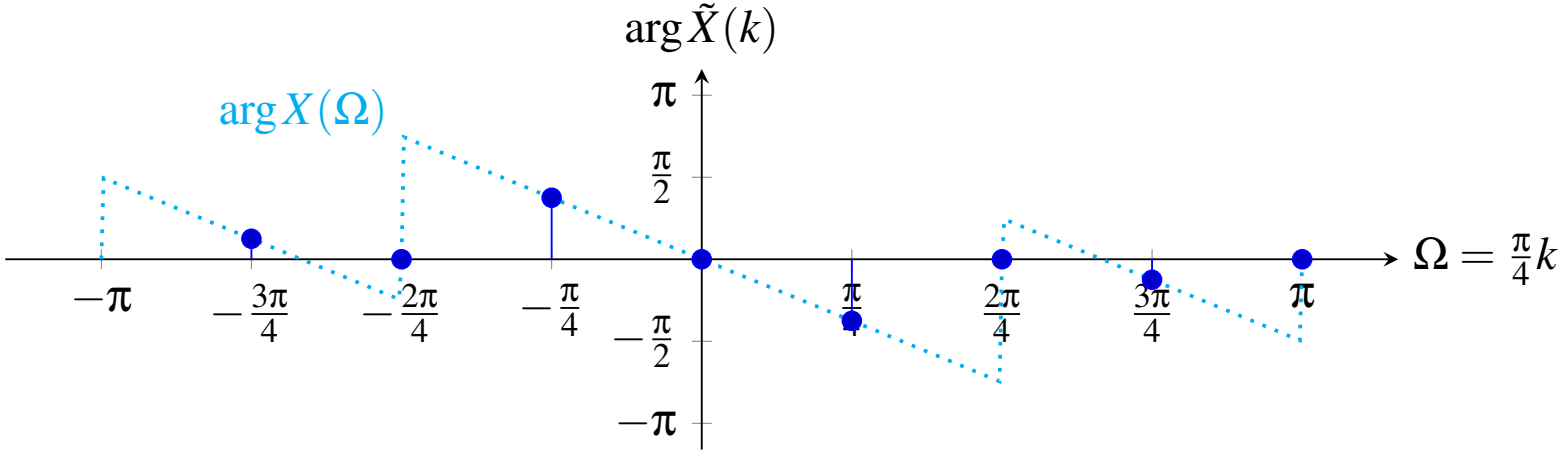


Phase Spectrum

Spectral Sampling Example: $N = 8$

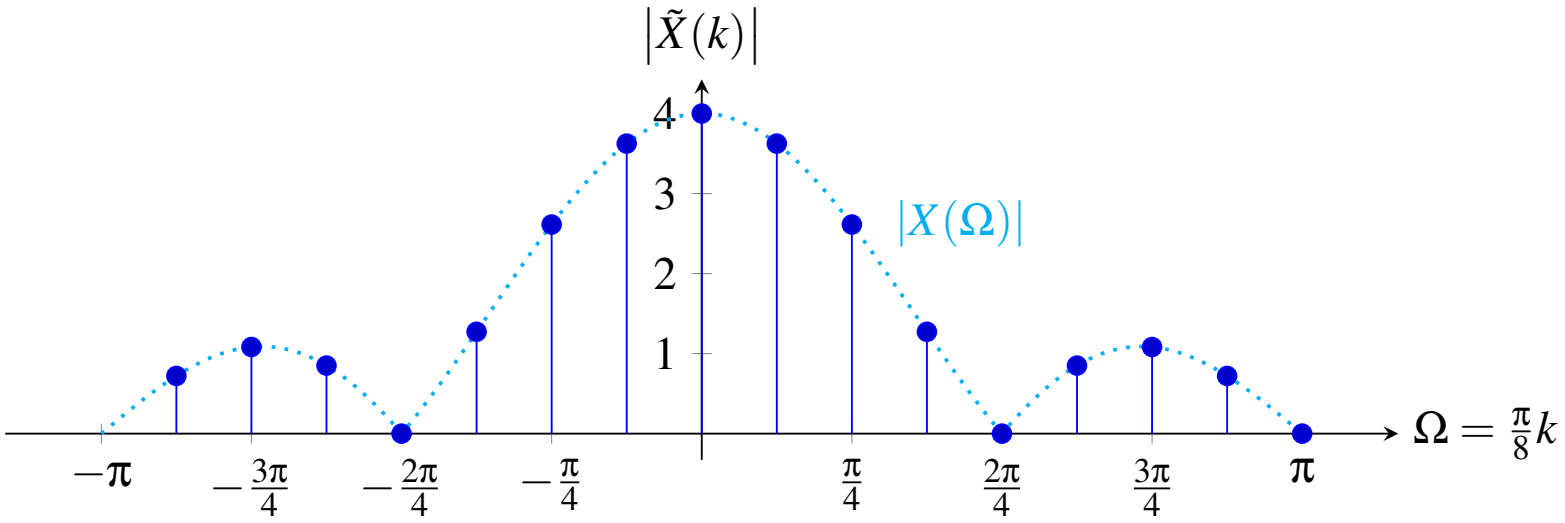


Magnitude Spectrum

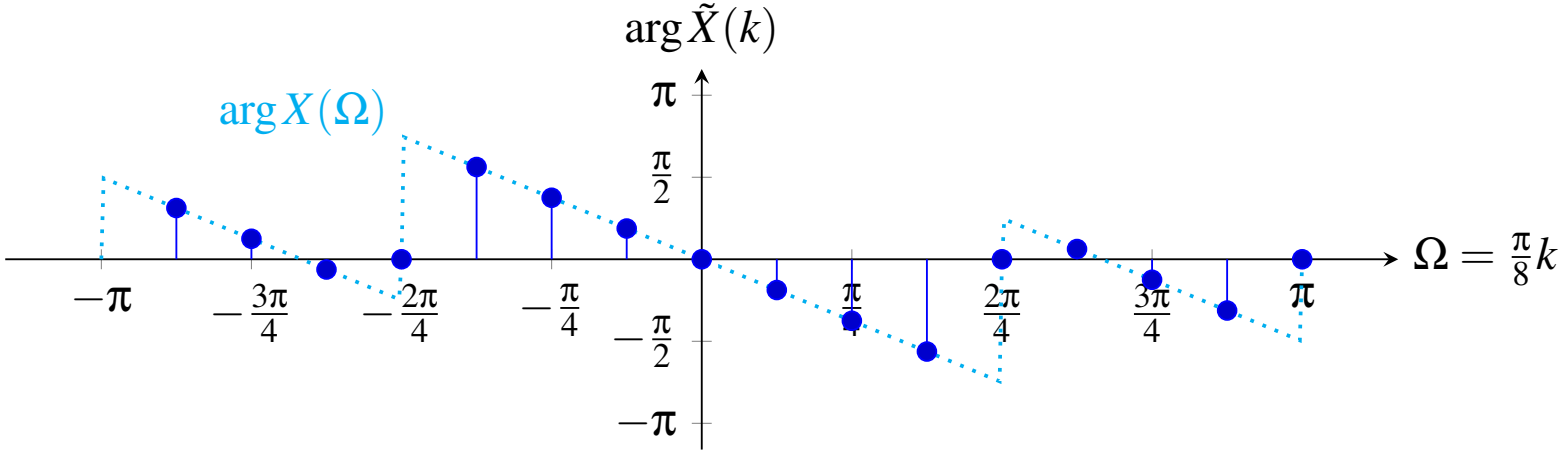


Phase Spectrum

Spectral Sampling Example: $N = 16$

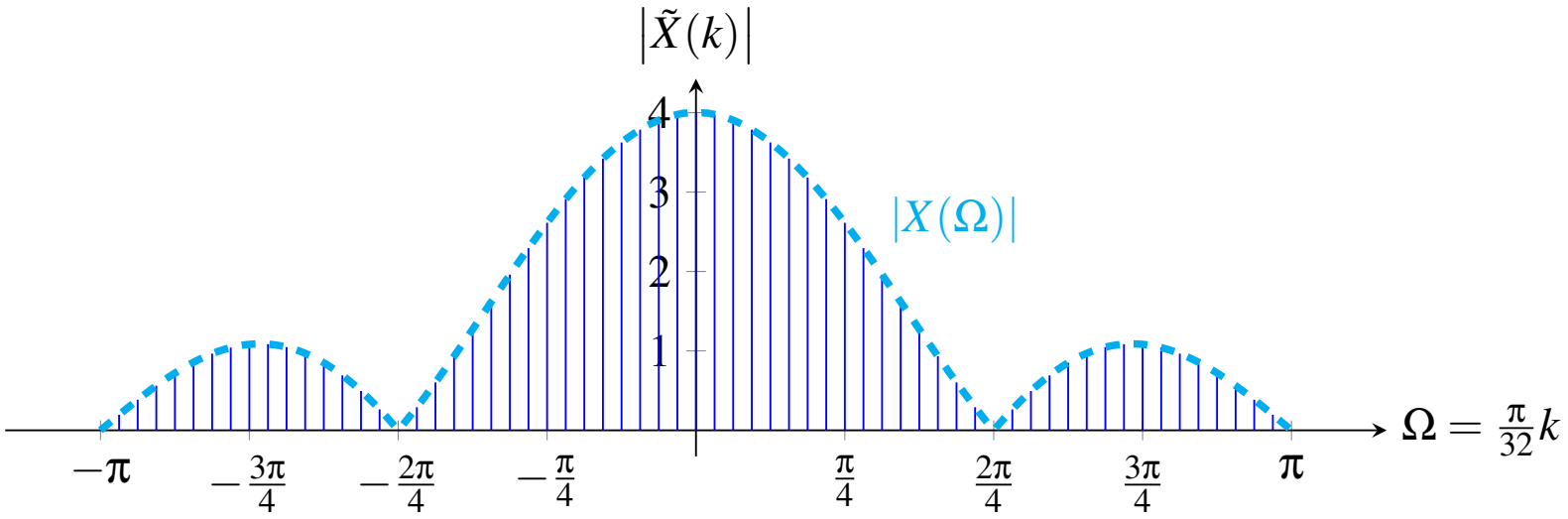


Magnitude Spectrum

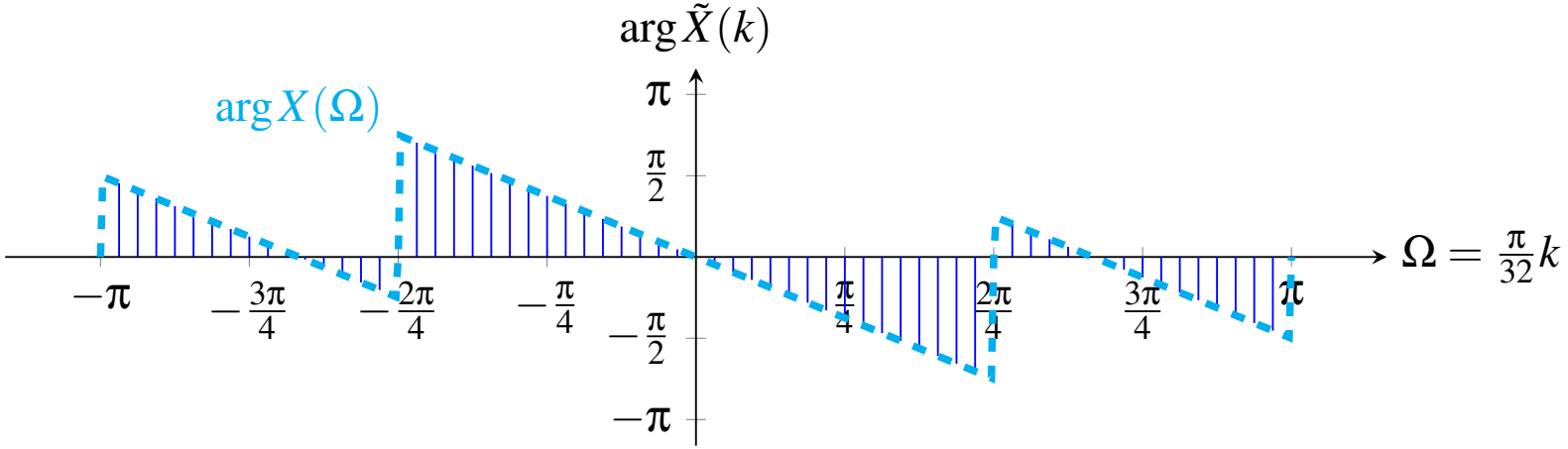


Phase Spectrum

Spectral Sampling Example: $N = 64$



Magnitude Spectrum



Phase Spectrum

Section 11.8

Application: Filtering

Filtering

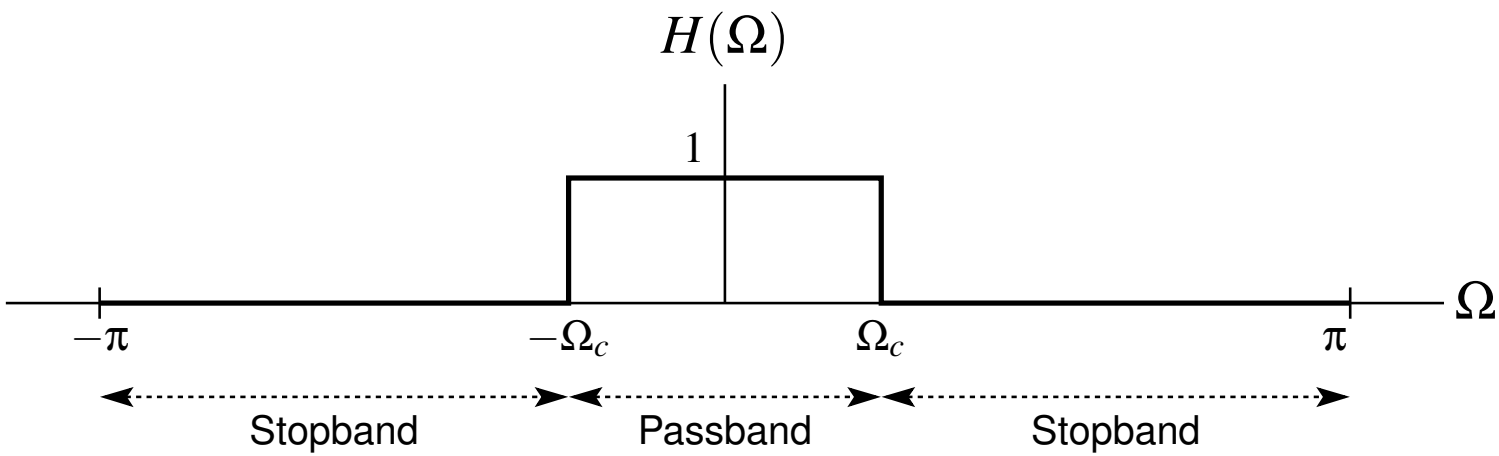
- In many applications, we want to ***modify the spectrum*** of a signal by either amplifying or attenuating certain frequency components.
- This process of modifying the frequency spectrum of a signal is called **filtering**.
- A system that performs a filtering operation is called a **filter**.
- Many types of filters exist.
- **Frequency selective filters** pass some frequencies with little or no distortion, while significantly attenuating other frequencies.
- Several basic types of frequency-selective filters include: lowpass, highpass, and bandpass.

Ideal Lowpass Filter

- An **ideal lowpass filter** eliminates all baseband frequency components with a frequency whose magnitude is greater than some cutoff frequency, while leaving the remaining baseband frequency components unaffected.
- Such a filter has a ***frequency response*** H of the form

$$H(\Omega) = \begin{cases} 1 & |\Omega| \leq \Omega_c \\ 0 & \Omega_c < |\Omega| \leq \pi, \end{cases}$$

  where Ω_c is the **cutoff frequency**.
- A plot of this frequency response is given below.

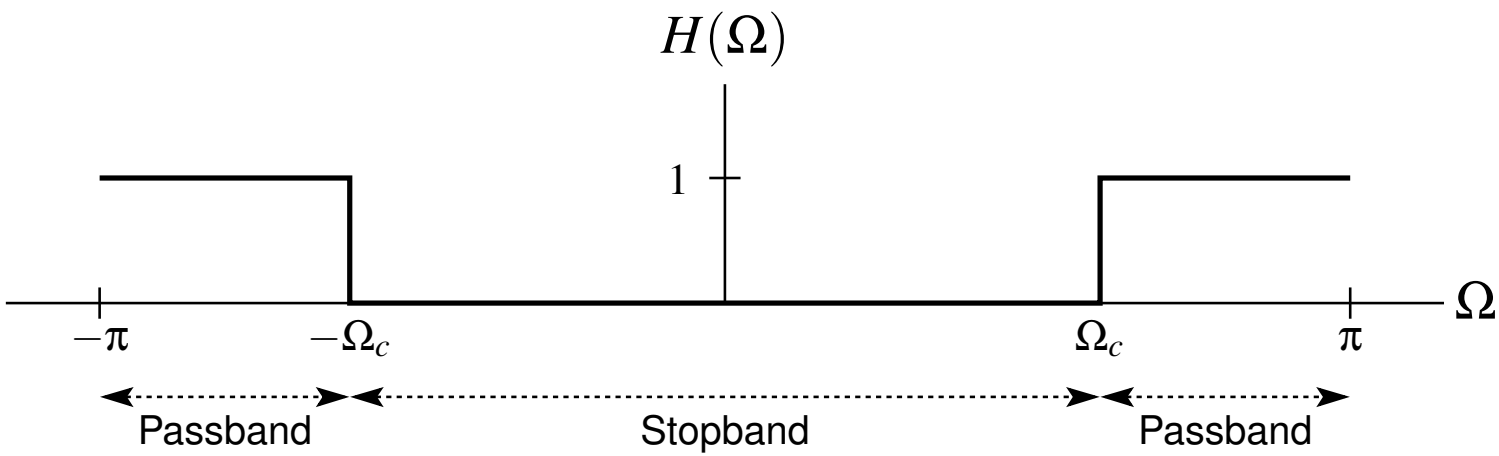


Ideal Highpass Filter

- An **ideal highpass filter** eliminates all baseband frequency components with a frequency whose magnitude is less than some cutoff frequency, while leaving the remaining baseband frequency components unaffected.
- Such a filter has a ***frequency response*** H of the form

$$H(\Omega) = \begin{cases} 1 & \Omega_c < |\Omega| \leq \pi \\ 0 & |\Omega| \leq \Omega_c, \end{cases}$$

  where Ω_c is the **cutoff frequency**.
- A plot of this frequency response is given below.

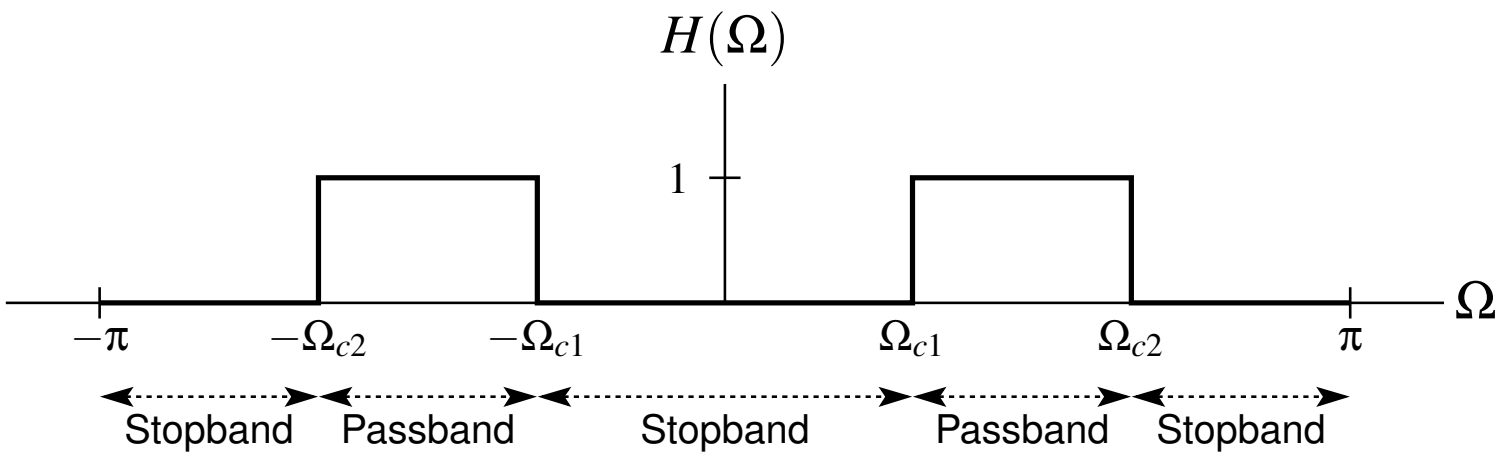


Ideal Bandpass Filter

- An **ideal bandpass filter** eliminates all baseband frequency components with a frequency whose magnitude does not lie in a particular range, while leaving the remaining baseband frequency components unaffected.
- Such a filter has a ***frequency response*** H of the form

$$H(\Omega) = \begin{cases} 1 & \Omega_{c1} \leq |\Omega| \leq \Omega_{c2} \\ 0 & |\Omega| < \Omega_{c1} \text{ or } \Omega_{c2} < |\Omega| < \pi, \end{cases}$$

  where the limits of the passband are Ω_{c1} and Ω_{c2}.
- A plot of this frequency response is given below.



Part 12

z Transform (ZT)

Motivation Behind the z Transform

- Another important mathematical tool in the study of signals and systems is known as the z transform.
- The z transform can be viewed as a ***generalization of the (classical) Fourier transform***.
- Due to its more general nature, the z transform has a number of ***advantages*** over the (classical) Fourier transform.
- First, the z transform representation ***exists for some sequences that do not have a Fourier transform representation***. So, we can handle some sequences with the z transform that cannot be handled with the Fourier transform.
- Second, since the z transform is a more general tool, it can provide ***additional insights*** beyond those facilitated by the Fourier transform.

Motivation Behind the z Transform (Continued)

- Earlier, we saw that complex exponentials are eigensequences of LTI systems.
- In particular, for a LTI system $\mathcal{H}$ with impulse response h, we have that

$$\mathcal{H}\{z^n\}(n) = H(z)z^n \quad \text{where} \quad H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}.$$

- Previously, we referred to H as the system function.
- As it turns out, H is the z transform of h.
- Since the z transform has already appeared earlier in the context of LTI systems, it is clearly a useful tool.
- Furthermore, as we will see, the z transform has many additional uses.

Section 12.1

z Transform

(Bilateral) z Transform

- The (bilateral) **z transform** of the sequence x, denoted $\mathcal{Z}x$ or X, is defined as

$$\mathcal{Z}x(z) = X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}.$$

- The **inverse z transform** of X, denoted $\mathcal{Z}^{-1}X$ or x, is then given by

$$\mathcal{Z}^{-1}X(n) = x(n) = \frac{1}{2\pi j}\oint_{\Gamma} X(z)z^{n-1}dz,$$

  where Γ is a counterclockwise closed circular contour centered at the origin and with radius r such that Γ is in the ROC of X.

- We refer to x and X as a **z transform pair** and denote this relationship as

$$x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z).$$

- In practice, we do not usually compute the inverse z transform by directly using the formula from above. Instead, we resort to other means (to be discussed later).

Bilateral and Unilateral z Transform

- Two different versions of the z transform are commonly used:
  1. the ***bilateral*** (or ***two-sided***) z transform; and
  2. the ***unilateral*** (or ***one-sided***) z transform.
- The unilateral z transform is most frequently used to solve systems of linear difference equations with nonzero initial conditions.
- As it turns out, the only difference between the definitions of the bilateral and unilateral z transforms is in the ***lower limit of summation***.
- In the bilateral case, the lower limit is $-\infty$, whereas in the unilateral case, the lower limit is 0.
- For the most part, we will focus our attention primarily on the bilateral z transform.
- We will, however, briefly introduce the unilateral z transform as a tool for solving difference equations.
- Unless otherwise noted, all subsequent references to the z transform should be understood to mean ***bilateral*** z transform.

Relationship Between Z and Fourier Transforms

- Let X and X_{F} denote the z and (DT) Fourier transforms of x, respectively.
- The function $X(z)$ evaluated at $z = e^{j\Omega}$ (where Ω is real) yields $X_{\mathsf{F}}(\Omega)$. That is,

$$X(e^{j\Omega}) = X_{\mathsf{F}}(\Omega).$$

- Due to the preceding relationship, the Fourier transform of x is sometimes written as $X(e^{j\Omega})$.
- The function $X(z)$ evaluated at an arbitrary complex value $z = re^{j\Omega}$ (where $r = |z|$ and $\Omega = \arg z$) can also be expressed in terms of a Fourier transform involving x. In particular, we have

$$X(re^{j\Omega}) = X'_{\mathsf{F}}(\Omega),$$

  where X'_{F} is the (DT) Fourier transform of $x'(n) = r^{-n}x(n)$.
- So, in general, the z transform of x is the Fourier transform of an exponentially-weighted version of x.
- Due to this weighting, the z transform of a sequence may exist when the Fourier transform of the same sequence does not.

THIS SLIDE IS INTENTIONALLY LEFT BLANK.

Section 12.2

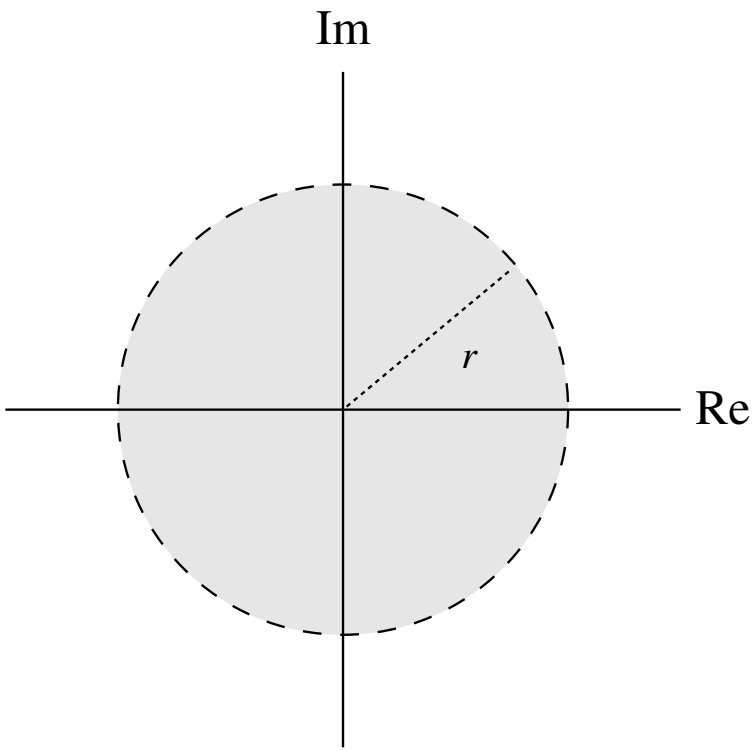
Region of Convergence (ROC)

Disk

- A **disk** with center 0 and radius r is the set of all complex numbers z satisfying

$$|z| < r,$$

where r is a real constant and $r > 0$.

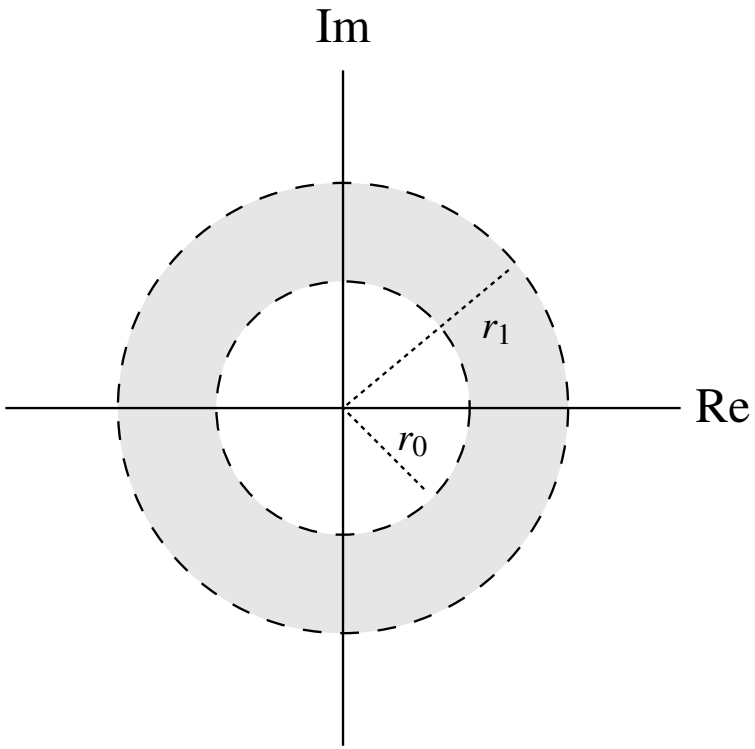


Annulus

- An **annulus** with center 0, inner radius r_0, and outer radius r_1 is the set of all complex numbers z satisfying

$$r_0 < |z| < r_1,$$

where r_0 and r_1 are real constants and $0 < r_0 < r_1$.

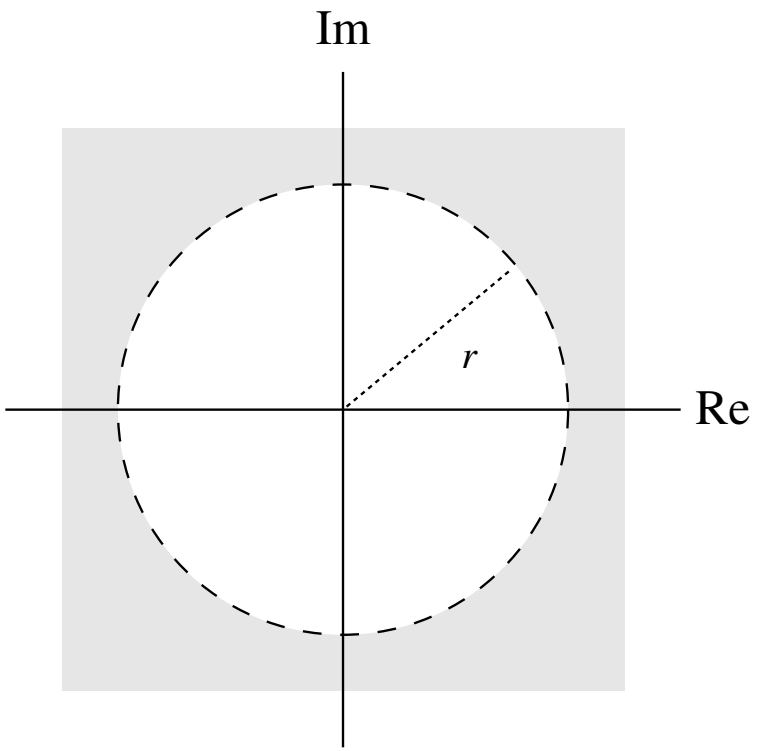


Circle Exterior

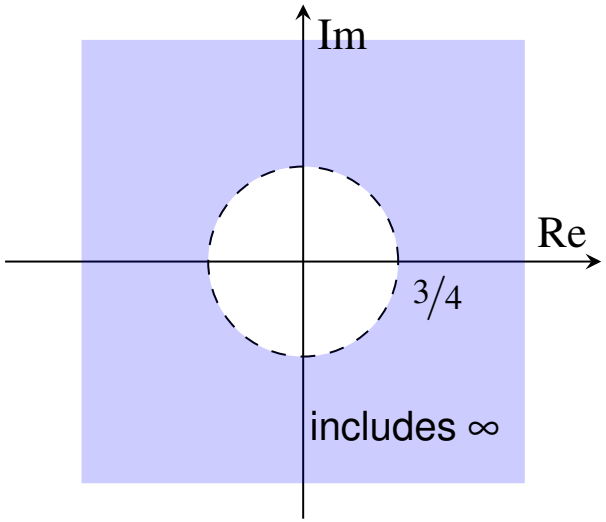
- The **exterior of a circle** with center 0 and radius r is the set of all complex numbers z satisfying

$$|z| > r,$$

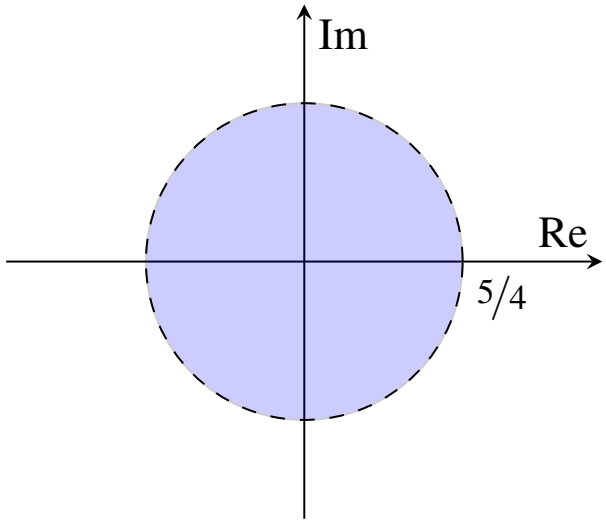
where r is a real constant and $r > 0$.



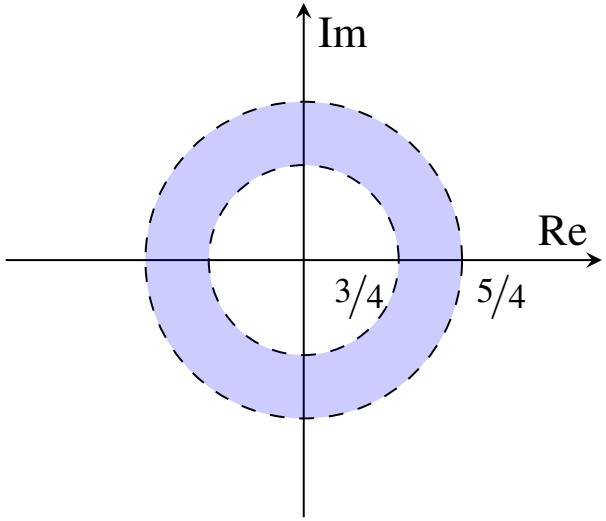
Example: Set Intersection



R_1

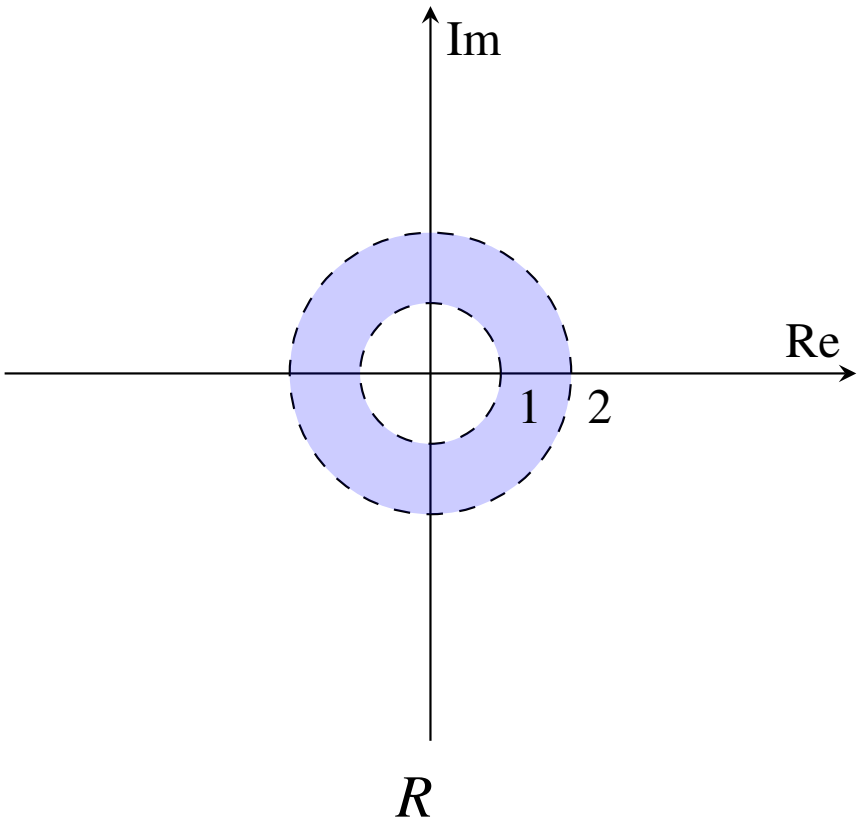


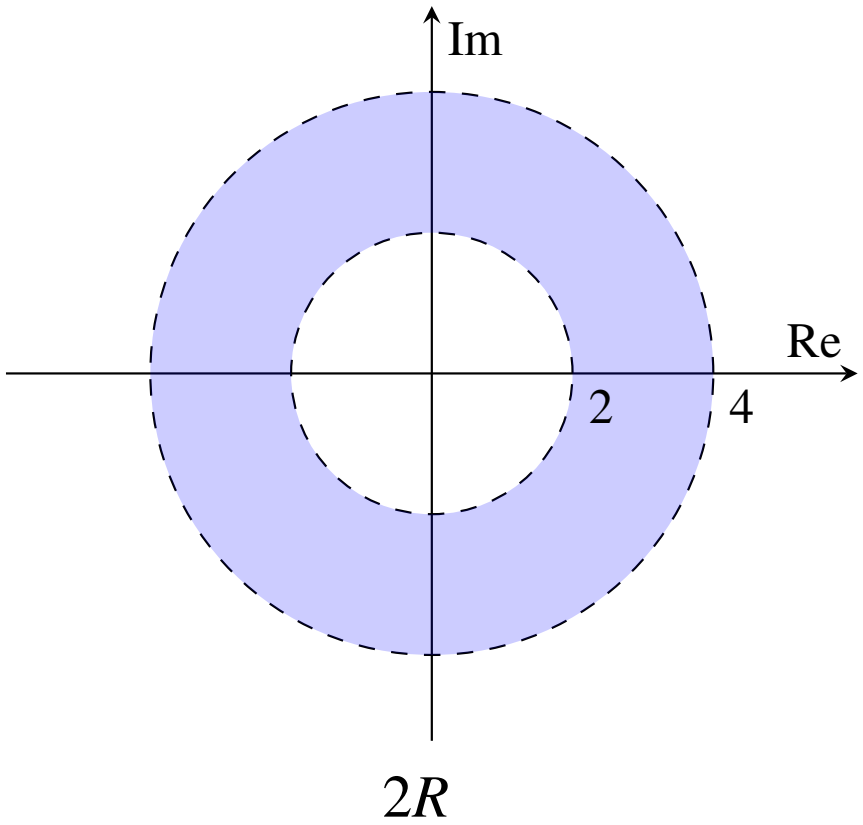
R_2



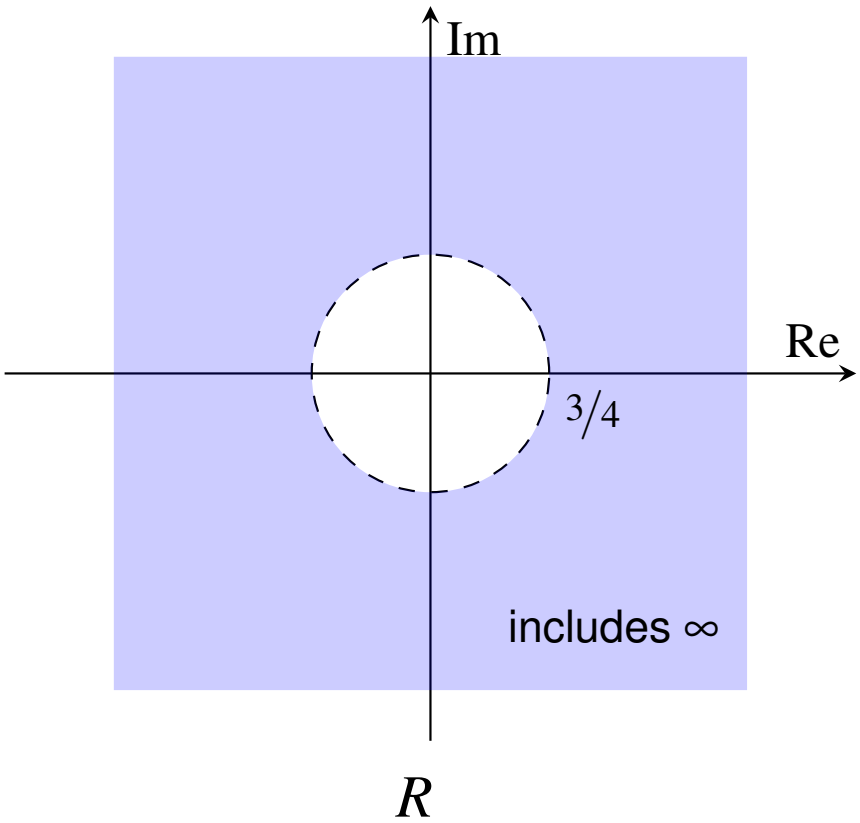
$R_1 \cap R_2$

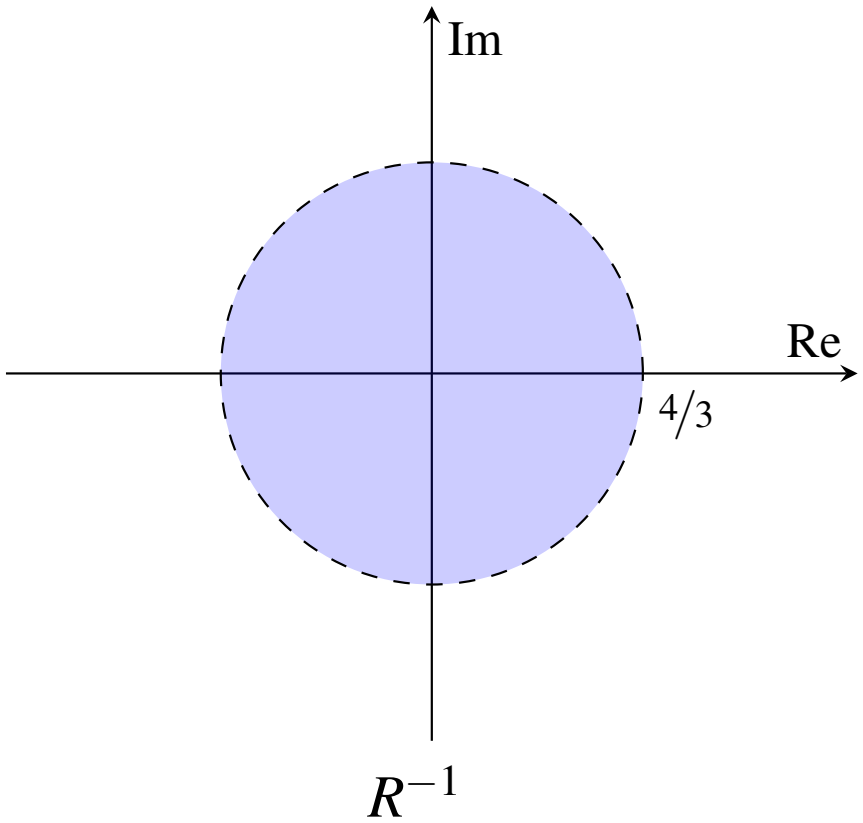
Example: Scalar Multiple of a Set





Example: Reciprocal of a Set



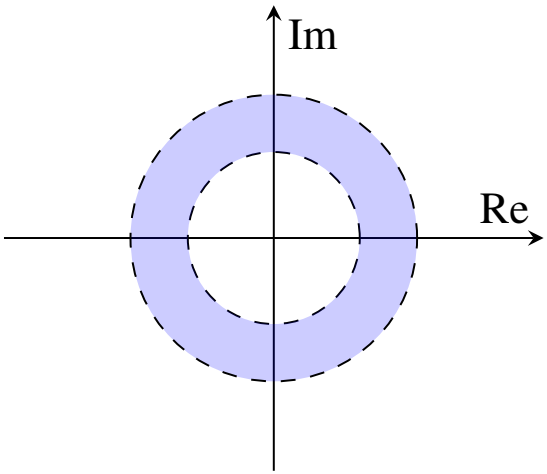


Region of Convergence (ROC)

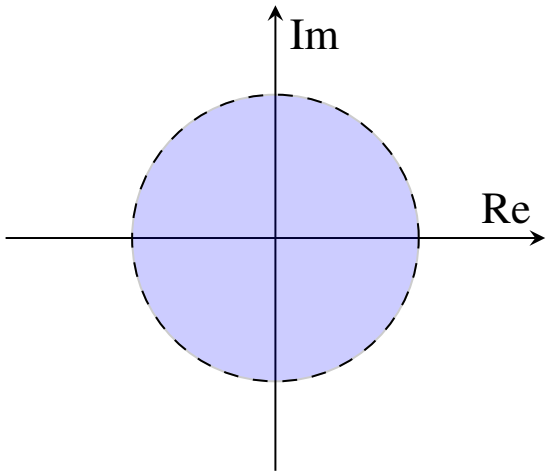
- As we saw earlier, for a sequence x, the complete specification of its z transform X requires not only an algebraic expression for X, but also the ROC associated with X.
- Two very different sequences can have the same algebraic expressions for X.
- Now, we examine some of the constraints on the ROC (of the z transform) for various classes of sequences.

Property 1: General Form

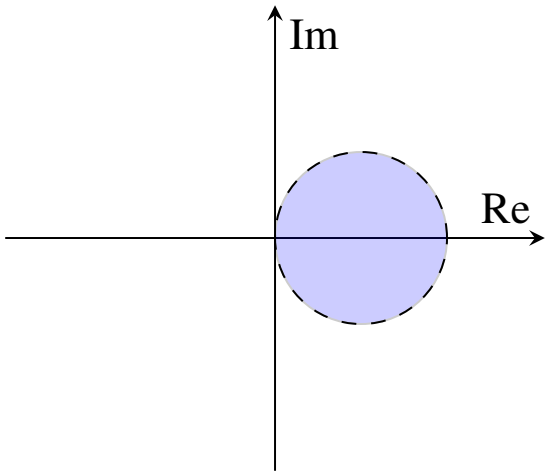
- The ROC of a z transform consists of ***concentric circles centered at*** 0 in the complex plane.
- That is, if a point z_0 is in the ROC, then the circle centered at 0 passing through z_0 (i.e., $|z| = |z_0|$) is also in the ROC.
- Some examples of sets that would be either valid or invalid as ROCs are shown below.



Valid



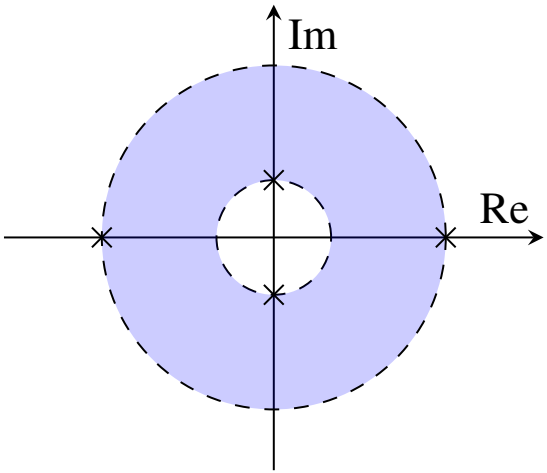
Valid



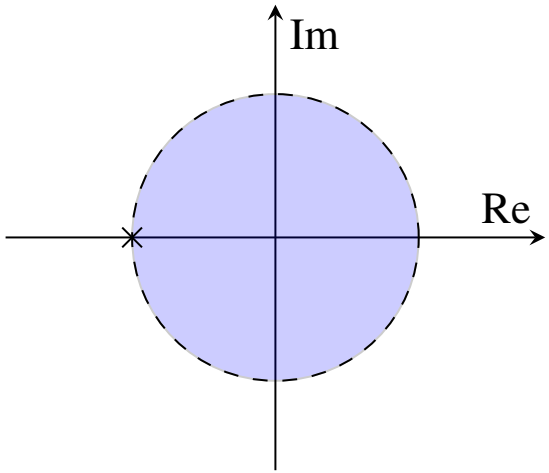
Invalid

Property 2: Rational z Transforms

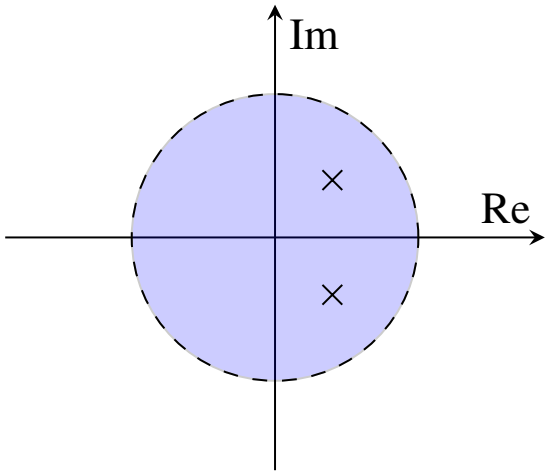
- If a z transform X is a ***rational*** function, then the ROC of X ***does not contain any poles*** and is ***bounded by poles or extends to infinity***.
- Some examples of sets that would be either valid or invalid as ROCs of rational z transforms are shown below.



Valid



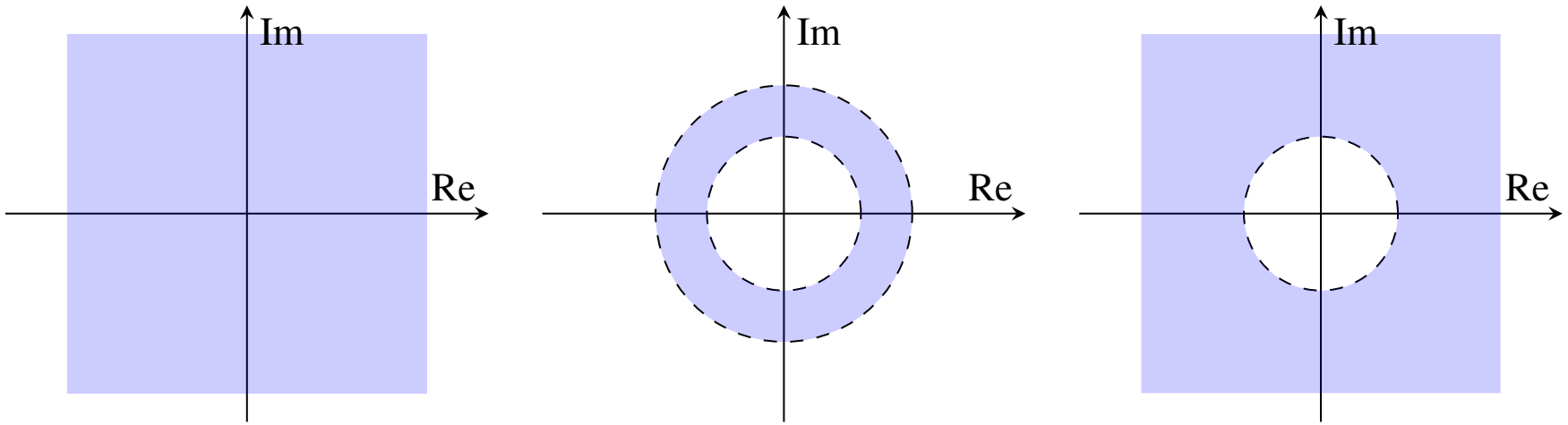
Valid



Invalid

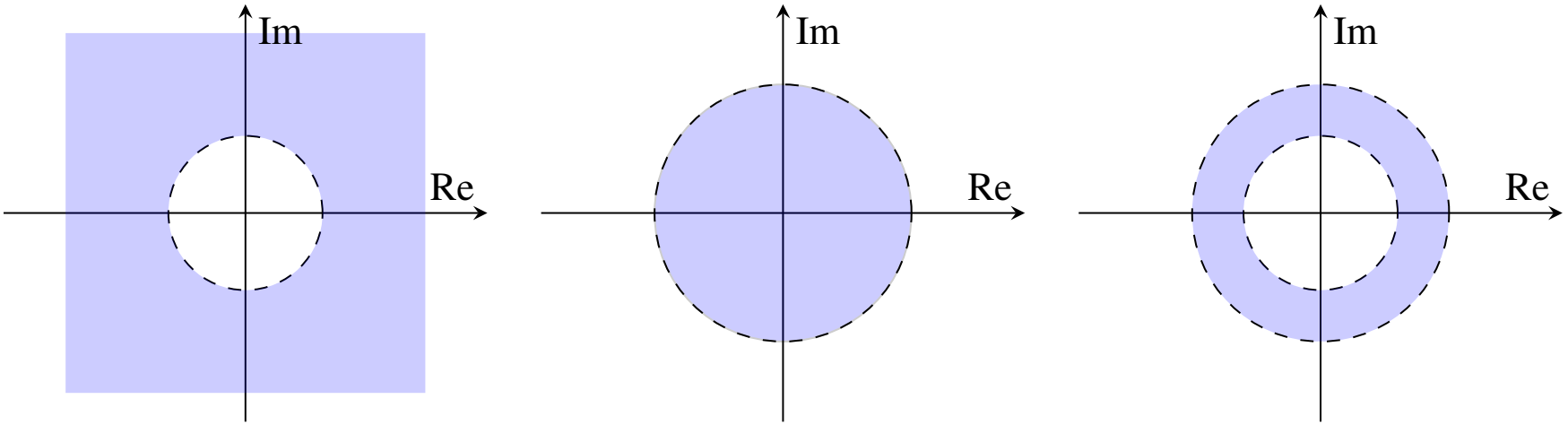
Property 3: Finite-Duration Sequences

- If a sequence x is ***finite duration*** and its z transform X converges for at least one point, then X converges for ***all points*** the complex plane, ***except possibly*** 0 ***and/or*** ∞.
- Some examples of sets that would be either valid or invalid as ROCs for X, if x is finite duration, are shown below.



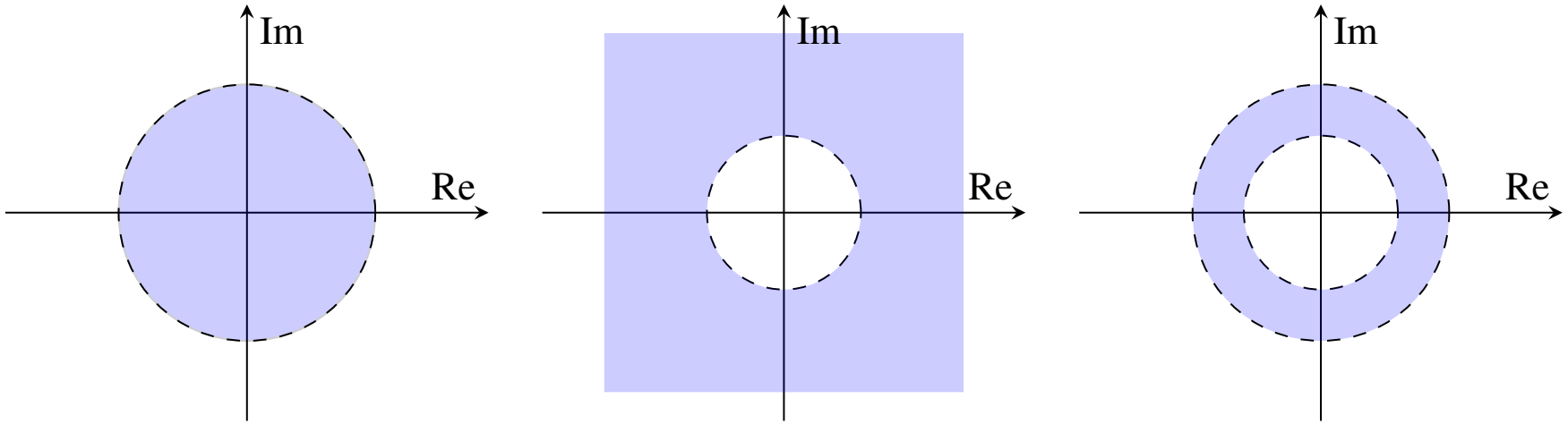
Property 4: Right-Sided Sequences

- If a sequence x is ***right sided*** and the circle $|z| = r_0$ is in the ROC of $X = \mathcal{Z}x$, then all (finite) values of z for which $|z| > r_0$ will also be in the ROC of X (i.e., the ROC contains the exterior of a circle centered at 0, possibly including ∞).
- Thus, if x is ***right sided but not left sided***, the ROC of X is the ***exterior of a circle centered at*** 0, possibly including ∞.
- Examples of sets that would be either valid or invalid as ROCs for X, if x is right sided but not left sided, are shown below.



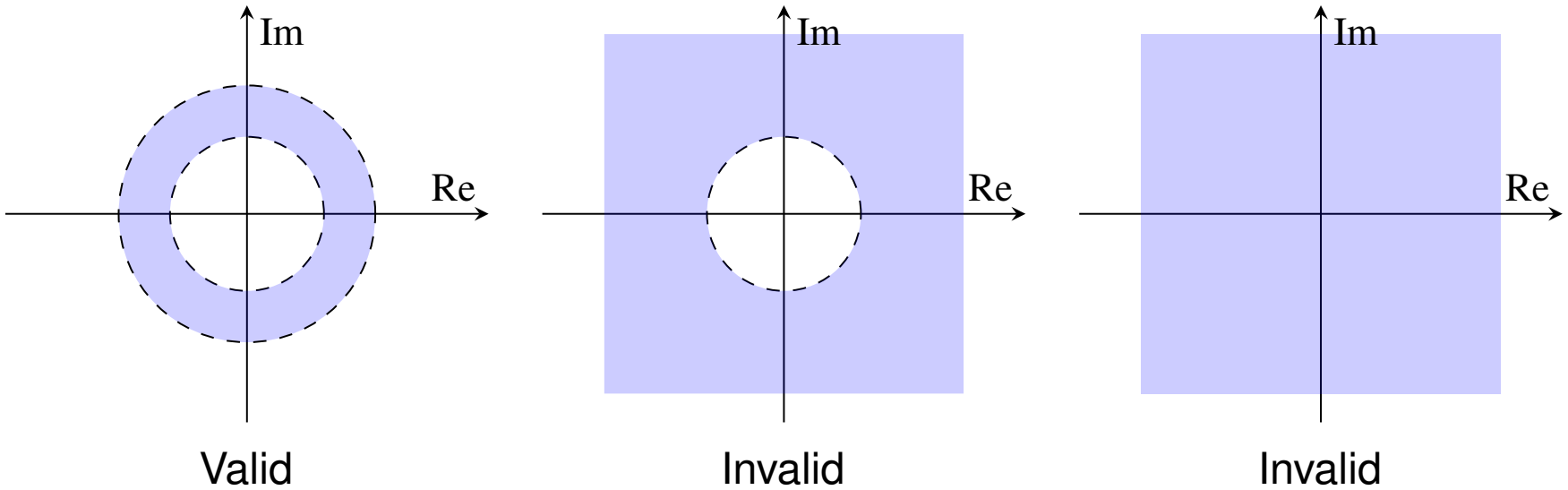
Property 5: Left-Sided Sequences

- If a sequence x is ***left sided*** and the circle $|z| = r_0$ is in the ROC of $X = \mathcal{Z}x$, then all values of z for which $0 < |z| < r_0$ will also be in the ROC of X (i.e., the ROC contains a disk centered at 0, possibly excluding 0).
- Thus, if x is ***left sided but not right sided***, the ROC of X is a ***disk centered at*** 0, possibly excluding 0.
- Examples of sets that would be either valid or invalid as ROCs for X, if x is left sided but not right sided, are shown below.



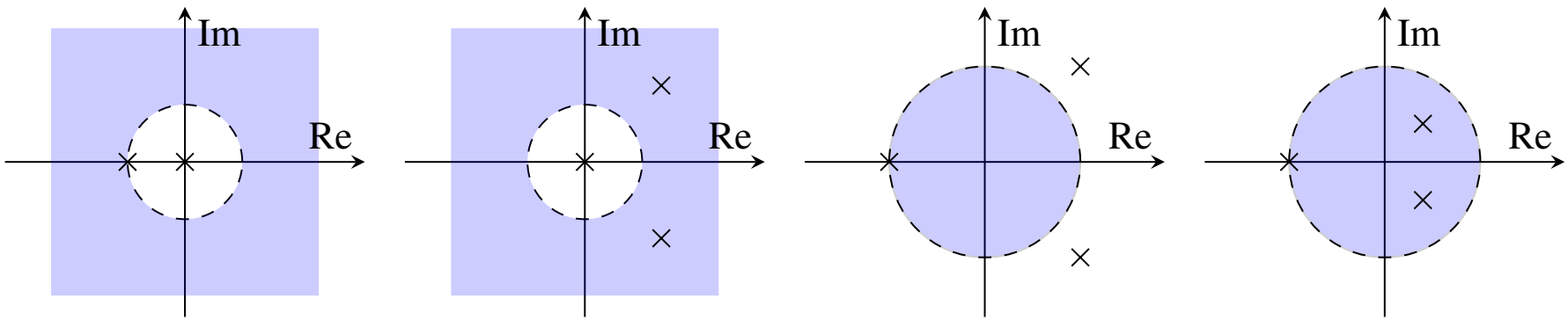
Property 6: Two-Sided Sequences

- If a sequence x is ***two sided*** and the circle $|z| = r_0$ is in the ROC of $X = \mathcal{Z}x$, then the ROC of X will consist of a ring that contains this circle (i.e., the ROC is an ***annulus centered at*** 0).
- Examples of sets that would be either valid or invalid as ROCs for X, if x is two sided, are shown below.



Property 7: More on Rational z Transforms

- If a sequence x has a ***rational*** z transform X (with at least one pole), then:
  1. If x is ***right sided***, then the ROC of X is the region outside the circle of radius equal to the largest magnitude of the poles of X (i.e., ***outside the outermost pole***), possibly including ∞.
  2. If x is ***left sided***, then the ROC of X is the region inside the circle of radius equal to the smallest magnitude of the nonzero poles of X and extending inward to, and possibly including, 0 (i.e., ***inside the innermost nonzero pole***).
- This property is implied by properties 1, 2, 4, and 5.
- Some examples of sets that would be either valid or invalid as ROCs for X, if X is rational and x is left/right sided, are given below.

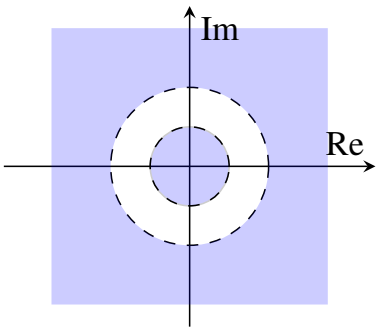


General Form of the ROC

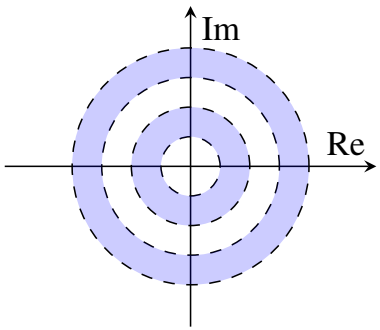
- To summarize the results of properties 3, 4, 5, and 6, if the z transform X of the sequence x exists, the ROC of X depends on the left- and right-sidedness of x as follows:

x		ROC of X
left sided	right sided	
yes	yes	everywhere, except possibly 0 and/or ∞
no	yes	exterior of circle centered at 0, possibly including ∞
yes	no	disk centered at 0, possibly excluding 0
no	no	annulus centered at 0

- Thus, we can infer that, if X exists, the ROC can only be of one of the forms listed above.
- For example, the sets shown below would not be valid as ROCs.



Invalid



Invalid

Section 12.3

Properties of the z Transform

Properties of the z Transform

Property	Time Domain	Z Domain	ROC
Linearity	$a_1x_1(n)+a_2x_2(n)$	$a_1X_1(z)+a_2X_2(z)$	At least $R_1 \cap R_2$
Translation	$x(n-n_0)$	$z^{-n_0}X(z)$	R except possible addition/deletion of 0
Modulation	$a^n x(n)$	$X(a^{-1}z)$	$\lvert a\rvert R$
Conjugation	$x^*(n)$	$X^*(z^*)$	R
Time Reversal	$x(-n)$	$X(1/z)$	R^{-1}
Upsampling	$(\uparrow M)x(n)$	$X(z^M)$	$R^{1/M}$
Downsampling	$(\downarrow M)x(n)$	$\frac{1}{M}\sum_{k=0}^{M-1} X\left(e^{-j2\pi k/M} z^{1/M}\right)$	R^M
Convolution	$x_1 * x_2(n)$	$X_1(z)X_2(z)$	At least $R_1 \cap R_2$
Z-Domain Diff.	$nx(n)$	$-z\frac{d}{dz}X(z)$	R
Differencing	$x(n)-x(n-1)$	$(1-z^{-1})X(z)$	At least $R \cap \lvert z\rvert > 0$
Accumulation	$\sum_{k=-\infty}^{n} x(k)$	$\frac{z}{z-1}X(z)$	At least $R \cap \lvert z\rvert > 1$

Property	
Initial Value Theorem	$x(0) = \lim_{z\to\infty} X(z)$
Final Value Theorem	$\lim_{n\to\infty} x(n) = \lim_{z\to 1}[(z-1)X(z)]$

z Transform Pairs

Pair	$x(n)$	$X(z)$	ROC
1	$\delta(n)$	1	All z
2	$u(n)$	$\frac{z}{z-1} = \frac{1}{1-z^{-1}}$	$\lvert z\rvert > 1$
3	$-u(-n-1)$	$\frac{z}{z-1} = \frac{1}{1-z^{-1}}$	$\lvert z\rvert < 1$
4	$nu(n)$	$\frac{z}{(z-1)^2} = \frac{z^{-1}}{(1-z^{-1})^2}$	$\lvert z\rvert > 1$
5	$-nu(-n-1)$	$\frac{z}{(z-1)^2} = \frac{z^{-1}}{(1-z^{-1})^2}$	$\lvert z\rvert < 1$
6	$a^n u(n)$	$\frac{z}{z-a} = \frac{1}{1-az^{-1}}$	$\lvert z\rvert > \lvert a\rvert$
7	$-a^n u(-n-1)$	$\frac{z}{z-a} = \frac{1}{1-az^{-1}}$	$\lvert z\rvert < \lvert a\rvert$
8	$na^n u(n)$	$\frac{az}{(z-a)^2} = \frac{az^{-1}}{(1-az^{-1})^2}$	$\lvert z\rvert > \lvert a\rvert$
9	$-na^n u(-n-1)$	$\frac{az}{(z-a)^2} = \frac{az^{-1}}{(1-az^{-1})^2}$	$\lvert z\rvert < \lvert a\rvert$
10	$\frac{(n+1)(n+2)\cdots(n+m-1)}{(m-1)!}a^n u(n)$	$\frac{z^m}{(z-a)^m} = \frac{1}{(1-az^{-1})^m}$	$\lvert z\rvert > \lvert a\rvert$
11	$-\frac{(n+1)(n+2)\cdots(n+m-1)}{(m-1)!}a^n u(-n-1)$	$\frac{z^m}{(z-a)^m} = \frac{1}{(1-az^{-1})^m}$	$\lvert z\rvert < \lvert a\rvert$

z Transform Pairs (Continued)

Pair	$x(n)$	$X(z)$	ROC
12	$\cos(\Omega_0 n)u(n)$	$\frac{z(z-\cos\Omega_0)}{z^2-2z\cos\Omega_0+1} = \frac{1-(\cos\Omega_0)z^{-1}}{1-(2\cos\Omega_0)z^{-1}+z^{-2}}$	$\lvert z\rvert > 1$
13	$-\cos(\Omega_0 n)u(-n-1)$	$\frac{z(z-\cos\Omega_0)}{z^2-2z\cos\Omega_0+1} = \frac{1-(\cos\Omega_0)z^{-1}}{1-(2\cos\Omega_0)z^{-1}+z^{-2}}$	$\lvert z\rvert < 1$
14	$\sin(\Omega_0 n)u(n)$	$\frac{z\sin\Omega_0}{z^2-2z\cos\Omega_0+1} = \frac{(\sin\Omega_0)z^{-1}}{1-(2\cos\Omega_0)z^{-1}+z^{-2}}$	$\lvert z\rvert > 1$
15	$-\sin(\Omega_0 n)u(-n-1)$	$\frac{z\sin\Omega_0}{z^2-2z\cos\Omega_0+1} = \frac{(\sin\Omega_0)z^{-1}}{1-(2\cos\Omega_0)z^{-1}+z^{-2}}$	$\lvert z\rvert < 1$
16	$a^n\cos(\Omega_0 n)u(n)$	$\frac{z(z-a\cos\Omega_0)}{z^2-2az\cos\Omega_0+a^2} = \frac{1-(a\cos\Omega_0)z^{-1}}{1-(2a\cos\Omega_0)z^{-1}+a^2z^{-2}}$	$\lvert z\rvert > \lvert a\rvert$
17	$a^n\sin(\Omega_0 n)u(n)$	$\frac{az\sin\Omega_0}{z^2-2az\cos\Omega_0+a^2} = \frac{(a\sin\Omega_0)z^{-1}}{1-(2a\cos\Omega_0)z^{-1}+a^2z^{-2}}$	$\lvert z\rvert > \lvert a\rvert$
18	$u(n)-u(n-M), M>0$	$\frac{z(1-z^{-M})}{z-1} = \frac{1-z^{-M}}{1-z^{-1}}$	$\lvert z\rvert > 0$
19	$a^{\lvert n\rvert}, \lvert a\rvert < 1$	$\frac{(a-a^{-1})z}{(z-a)(z-a^{-1})}$	$\lvert a\rvert < \lvert z\rvert < \lvert a^{-1}\rvert$

Linearity

- If $x_1(n) \overset{\text{ZT}}{\longleftrightarrow} X_1(z)$ with ROC R_1 and $x_2(n) \overset{\text{ZT}}{\longleftrightarrow} X_2(z)$ with ROC R_2, then

  $$a_1 x_1(n) + a_2 x_2(n) \overset{\text{ZT}}{\longleftrightarrow} a_1 X_1(z) + a_2 X_2(z) \quad \text{with ROC } R \text{ containing } R_1 \cap R_2,$$

  where a_1 and a_2 are arbitrary complex constants.
- This is known as the **linearity property** of the z transform.
- The ROC always contains the intersection but could be larger (in the case that pole-zero cancellation occurs).

Translation (Time Shifting)

- If $x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z)$ with ROC R, then

$$x(n-n_0) \overset{\text{ZT}}{\longleftrightarrow} z^{-n_0}X(z) \text{ with ROC } R',$$

  where n_0 is an integer constant and R' is the same as R except for the possible addition or deletion of zero or infinity.

- This is known as the **translation (or time-shifting) property** of the z transform.

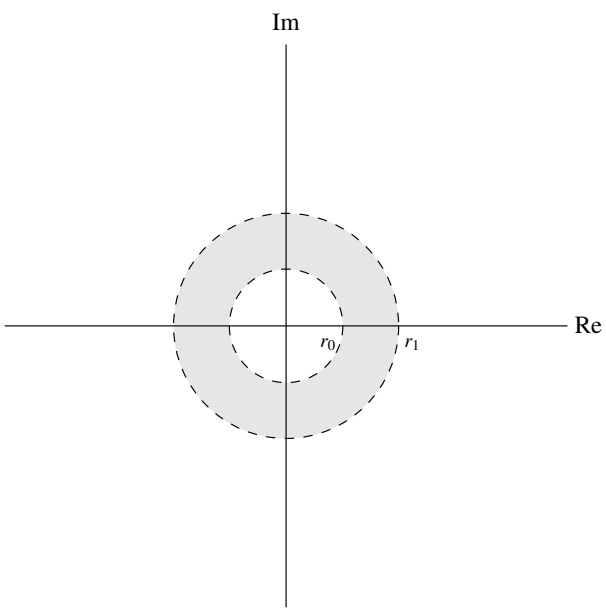
Z-Domain Scaling

- If $x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z)$ with ROC R, then

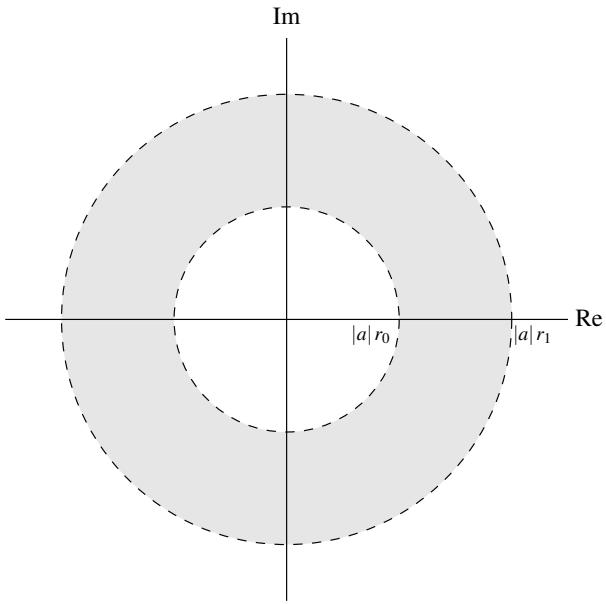
$$a^n x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z/a) \quad \text{with ROC } |a|R,$$

  where a is a nonzero constant.

- This is known as the **z-domain scaling property** of the z transform.
- As illustrated below, the ROC R is ***scaled*** by $|a|$.



R



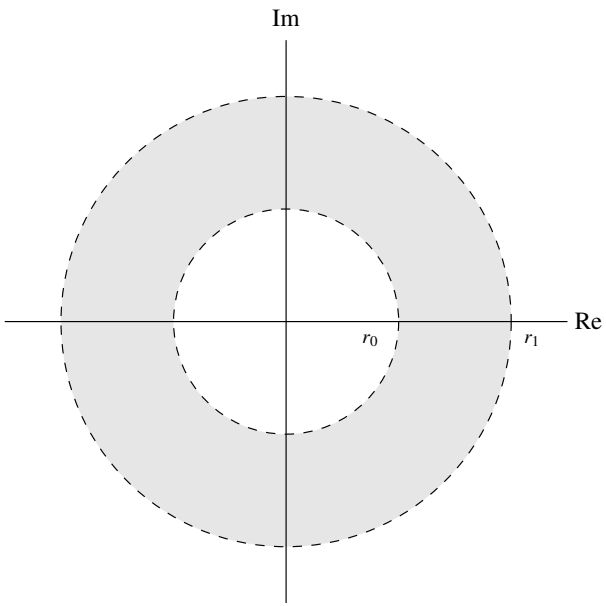
$|a|R$

Time Reversal

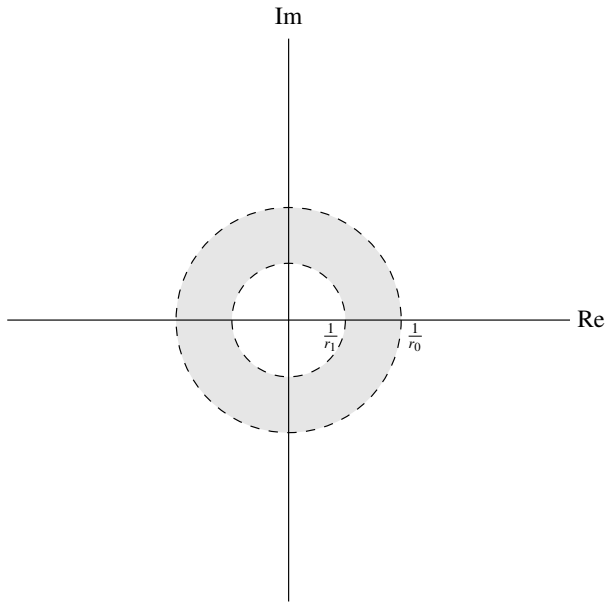
- If $x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z)$ with ROC R, then

$$x(-n) \overset{\text{ZT}}{\longleftrightarrow} X(1/z) \quad \text{with ROC } 1/R.$$

- This is known as the **time-reversal property** of the z transform.
- As illustrated below, the ROC R is ***reciprocated***.



R



$1/R$

Upsampling

- Define $(\uparrow M)x(n)$ as

$$(\uparrow M)x(n) = \begin{cases} x(n/M) & n/M \text{ is an integer} \\ 0 & \text{otherwise.} \end{cases}$$

- If $x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z)$ with ROC R, then

$$(\uparrow M)x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z^M) \quad \text{with ROC } R^{1/M}.$$

- This is known as the **upsampling (or time-expansion) property** of the z transform.

Downsampling

- If $x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z)$ with ROC R, then

$$(\downarrow M)x(n) \overset{\text{ZT}}{\longleftrightarrow} \frac{1}{M} \sum_{k=0}^{M-1} X\left(e^{-j2\pi k/M} z^{1/M}\right) \quad \text{with ROC } R^M.$$

- This is known as the **downsampling property** of the z transform.

Conjugation

- If $x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z)$ with ROC R, then

$$x^*(n) \overset{\text{ZT}}{\longleftrightarrow} X^*(z^*) \quad \text{with ROC } R.$$

- This is known as the **conjugation property** of the z transform.

Convolution

- If $x_1(n) \overset{\text{ZT}}{\longleftrightarrow} X_1(z)$ with ROC R_1 and $x_2(n) \overset{\text{ZT}}{\longleftrightarrow} X_2(z)$ with ROC R_2, then

$$x_1 * x_2(n) \overset{\text{ZT}}{\longleftrightarrow} X_1(z)X_2(z) \quad \text{with ROC containing } R_1 \cap R_2.$$

- This is known that the **convolution (or time-domain convolution) property** of the z transform.
- The ROC always contains the intersection but can be larger than the intersection (if pole-zero cancellation occurs).
- Convolution in the time domain becomes ***multiplication*** in the z domain.
- This can make dealing with LTI systems much easier in the z domain than in the time domain.

Z-Domain Differentiation

- If $x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z)$ with ROC R, then

$$nx(n) \overset{\text{ZT}}{\longleftrightarrow} -z\frac{d}{dz}X(z) \quad \text{with ROC } R.$$

- This is known as the **z-domain differentiation property** of the z transform.

Differencing

- If $x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z)$ with ROC R, then

$$x(n) - x(n-1) \overset{\text{ZT}}{\longleftrightarrow} (1 - z^{-1})X(z) \text{ for ROC containing } R \cap |z| > 0.$$

- This is known as the **differencing property** of the z transform.
- Differencing in the time domain becomes multiplication by $1 - z^{-1}$ in the z domain.
- This can make dealing with difference equations much easier in the z domain than in the time domain.

Accumulation

- If $x(n) \overset{\text{ZT}}{\longleftrightarrow} X(z)$ with ROC R, then

$$\sum_{k=-\infty}^{n} x(k) \overset{\text{ZT}}{\longleftrightarrow} \frac{z}{z-1} X(z) \text{ for ROC containing } R \cap |z| > 1.$$

- This is known as the **accumulation property** of the z transform.

Initial Value Theorem

- For a sequence x with z transform X, if x is causal, then

$$x(0) = \lim_{z \to \infty} X(z).$$

- This result is known as the **initial-value theorem**.

Final Value Theorem

- For a sequence x with z transform X, if x is causal and $\lim_{n\to\infty} x(n)$ exists, then

$$\lim_{n\to\infty} x(n) = \lim_{z\to 1}[(z-1)X(z)].$$

- This result is known as the **final-value theorem**.

More z Transform Examples

THIS SLIDE IS INTENTIONALLY LEFT BLANK.

Section 12.4

Determination of Inverse z Transform

Finding the Inverse z Transform

- Recall that the inverse z transform x of X is given by

$$x(n) = \frac{1}{2\pi j} \oint_{\Gamma} X(z) z^{n-1} dz,$$

  where Γ is a counterclockwise closed circular contour centered at the origin and with radius r such that Γ is in the ROC of X.
- Unfortunately, the above contour integration can often be ***quite tedious*** to compute.
- Consequently, we do not usually compute the inverse z transform directly using the above equation.
- For rational functions, the inverse z transform can be more easily computed using ***partial fraction expansions***.
- Using a partial fraction expansion, we can express a rational function as a sum of lower-order rational functions whose inverse z transforms can typically be found in tables.

Section 12.5

z Transform and LTI Systems

System Function of LTI Systems

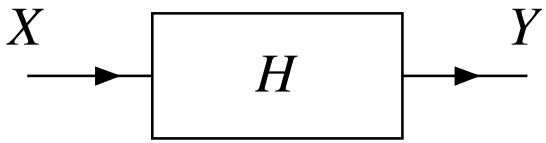
- Consider a LTI system with input x, output y, and impulse response h, and let X, Y, and H denote the z transforms of x, y, and h, respectively.
- Since $y(n) = x * h(n)$, the system is characterized in the z domain by

$$Y(z) = X(z)H(z).$$

- As a matter of terminology, we refer to H as the **system function** (or **transfer function**) of the system (i.e., the system function is the z transform of the impulse response).
- When viewed in the z domain, a LTI system forms its output by multiplying its input with its system function.
- A LTI system is completely characterized by its system function H.
- If the ROC of H includes the unit circle $|z| = 1$, then $H(e^{j\Omega})$ is the ***frequency response*** of the LTI system.

Block Diagram Representation of LTI Systems

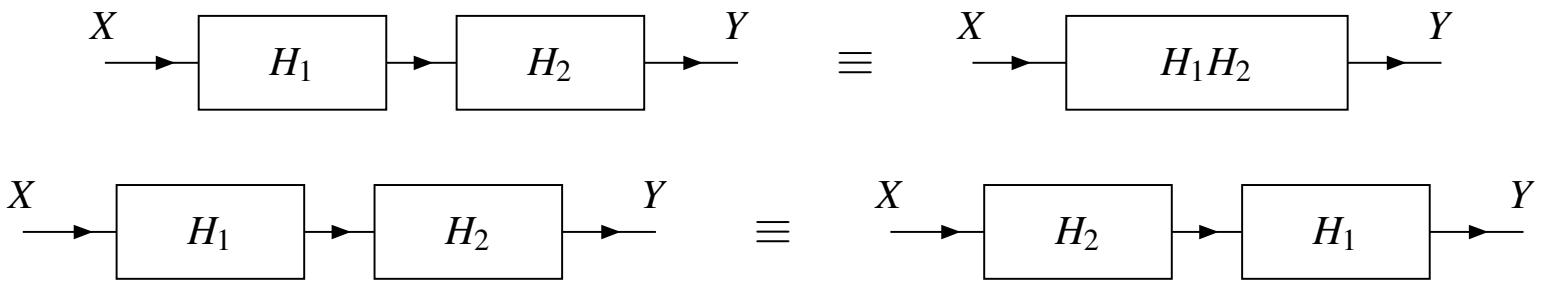
- Consider a LTI system with input x, output y, and impulse response h, and let X, Y, and H denote the z transforms of x, y, and h, respectively.
- Often, it is convenient to represent such a system in block diagram form in the z domain as shown below.



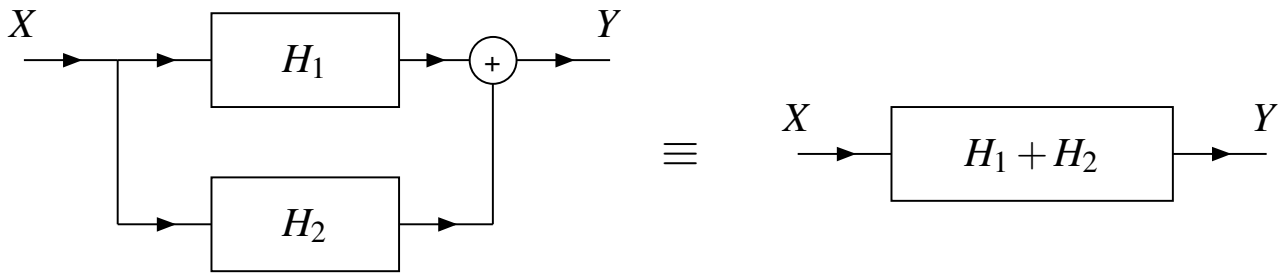
- Since a LTI system is completely characterized by its system function, we typically label the system with this quantity.

Interconnection of LTI Systems

- The ***series*** interconnection of the LTI systems with system functions H_1 and H_2 is the LTI system with system function $H = H_1 H_2$. That is, we have the equivalences shown below.



- The ***parallel*** interconnection of the LTI systems with impulse responses H_1 and H_2 is a LTI system with the system function $H = H_1 + H_2$. That is, we have the equivalence shown below.



Causality

- If a LTI system is ***causal***, its impulse response is causal, and therefore ***right sided***. From this, we have the result below.
- **Theorem.** A LTI system is ***causal*** if and only if the ROC of the system function is:
  1. the ***exterior of a circle, including*** ∞; or
  2. the ***entire complex plane, including*** ∞ and possibly excluding 0.
- **Theorem.** A LTI system with a ***rational*** system function H is causal if and only if:
  1. the ROC of H is the exterior of a (possibly degenerate) circle ***outside the outermost pole*** of H or, if H has no poles, the entire complex plane; and
  2. H is ***proper*** (i.e., when $H(z)$ is expressed as a ratio of polynomials in z, the order of the numerator polynomial does not exceed the order of the denominator polynomial).

BIBO Stability

- Whether or not a system is BIBO stable depends on the ROC of its system function.
- **Theorem.** A LTI system is ***BIBO stable*** if and only if the ROC of its system function contains the ***unit circle*** (i.e., $|z| = 1$).
- **Theorem.** A ***causal*** LTI system with a ***rational*** system function H is BIBO stable if and only if all of the poles of H lie inside the unit circle (i.e., each of the poles has a ***magnitude less than one***).

Invertibility

- A LTI system $\mathcal{H}$ with system function H is invertible if and only if there exists another LTI system with system function H_{inv} such that

$$H(z)H_{\text{inv}}(z) = 1,$$

  in which case H_{inv} is the system function of $\mathcal{H}^{-1}$ and

$$H_{\text{inv}}(z) = \frac{1}{H(z)}.$$

- Since distinct systems can have identical system functions (but with differing ROCs), the inverse of a LTI system is ***not necessarily unique***.
- In practice, however, we often desire a stable and/or causal system. So, although multiple inverse systems may exist, we are frequently only interested in ***one specific choice*** of inverse system (due to these additional constraints of stability and/or causality).

LTI Systems and Difference Equations

- Many LTI systems of practical interest can be represented using an ***Nth-order linear difference equation with constant coefficients***.
- Consider a system with input x and output y that is characterized by an equation of the form

$$\sum_{k=0}^{N} b_k y(n-k) = \sum_{k=0}^{M} a_k x(n-k) \quad \text{where} \quad M \leq N.$$

- Let h denote the impulse response of the system, and let X, Y, and H denote the z transforms of x, y, and h, respectively.
- One can show that $H(z)$ is given by

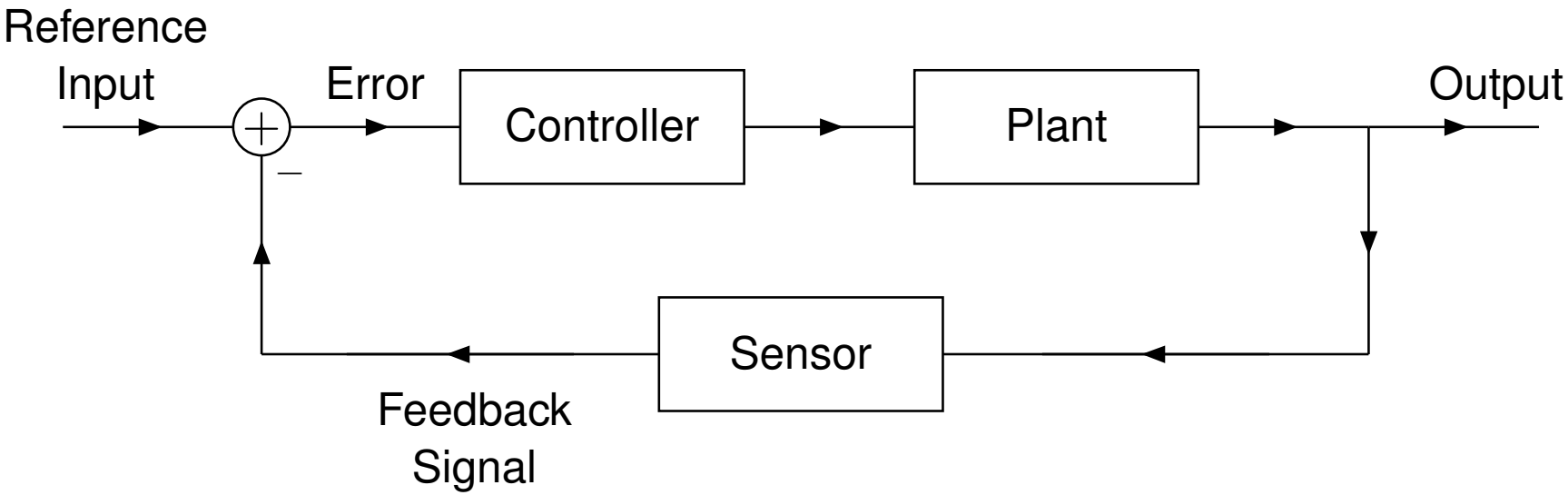
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} a_k z^{-k}}{\sum_{k=0}^{N} b_k z^{-k}}.$$

- Observe that, for a system of the form considered above, the system function is always ***rational***.

Section 12.6

Application: Analysis of Control Systems

Feedback Control Systems



- **input**: ***desired value*** of the quantity to be controlled
- **output**: ***actual value*** of the quantity to be controlled
- **error**: ***difference*** between the desired and actual values
- **plant**: system to be controlled
- **sensor**: device used to measure the actual output
- **controller**: device that monitors the error and changes the input of the plant with the goal of forcing the error to zero

Stability Analysis of Feedback Control Systems

- Often, we want to ensure that a system is BIBO stable.
- The BIBO stability property is more easily characterized in the z domain than in the time domain.
- Therefore, the z domain is extremely useful for the stability analysis of systems.

Section 12.7

Unilateral z Transform

Unilateral z Transform

- The **unilateral z transform** of the sequence x, denoted $\mathcal{Z}_{\mathrm{u}}x$ or X, is defined as

$$\mathcal{Z}_{\mathrm{u}}x(z) = X(z) = \sum_{n=0}^{\infty} x(n)z^{-n}.$$

- The unilateral z transform is related to the bilateral z transform as follows:

$$\mathcal{Z}_{\mathrm{u}}x(z) = \sum_{n=0}^{\infty} x(n)z^{-n} = \sum_{n=-\infty}^{\infty} x(n)u(n)z^{-n} = \mathcal{Z}\{xu\}(z).$$

- In other words, the unilateral z transform of the sequence x is simply the bilateral z transform of the sequence xu.
- Since $\mathcal{Z}_{\mathrm{u}}x = \mathcal{Z}\{xu\}$ and xu is always a ***right-sided*** sequence, the ROC associated with $\mathcal{Z}_{\mathrm{u}}x$ is always the ***exterior of a circle***.
- For this reason, we often ***do not explicitly indicate the ROC*** when working with the unilateral z transform.

Unilateral z Transform (Continued 1)

- With the unilateral z transform, the same inverse transform equation is used as in the bilateral case.
- The unilateral z transform is ***only invertible for causal sequences***. In particular, we have

$$\begin{aligned}
\mathcal{Z}_{\mathsf{u}}^{-1}\{\mathcal{Z}_{\mathsf{u}}\{x\}\}(n) &= \mathcal{Z}_{\mathsf{u}}^{-1}\{\mathcal{Z}\{xu\}\}(n) \\
&= \mathcal{Z}^{-1}\{\mathcal{Z}\{xu\}\}(n) \\
&= x(n)u(n) \\
&= \begin{cases} x(n) & n \geq 0 \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

- For a noncausal sequence x, we can only recover $x(n)$ for $n \geq 0$.

Unilateral z Transform (Continued 2)

- Due to the close relationship between the unilateral and bilateral z transforms, these two transforms have some similarities in their properties.
- Since these two transforms are not identical, however, their properties differ in some cases, often in subtle ways.

Properties of the Unilateral z Transform

Property	Time Domain	Z Domain
Linearity	$a_1x_1(n)+a_2x_2(n)$	$a_1X_1(z)+a_2X_2(z)$
Time Delay	$x(n-1)$	$z^{-1}X(z)+x(-1)$
Time Advance	$x(n+1)$	$zX(z)-zx(0)$
Modulation	$a^nx(n)$	$X(a^{-1}z)$
	$e^{j\Omega_0 n}x(n)$	$X(e^{-j\Omega_0}z)$
Conjugation	$x^*(n)$	$X^*(z^*)$
Upsampling	$(\uparrow M)x(n)$	$X(z^M)$
Downsampling	$(\downarrow M)x(n)$	$\frac{1}{M}\sum_{k=0}^{M-1}X\left(e^{-j2\pi k/M}z^{1/M}\right)$
Convolution	$x_1*x_2(n)$, x_1 and x_2 are causal	$X_1(z)X_2(z)$
Z-Domain Diff.	$nx(n)$	$-z\frac{d}{dz}X(z)$
Differencing	$x(n)-x(n-1)$	$(1-z^{-1})X(z)-x(-1)$
Accumulation	$\sum_{k=0}^{n}x(k)$	$\frac{1}{1-z^{-1}}X(z)$

Property	
Initial Value Theorem	$x(0)=\lim_{z\to\infty}X(z)$
Final Value Theorem	$\lim_{n\to\infty}x(n)=\lim_{z\to 1}[(z-1)X(z)]$

Unilateral z Transform Pairs

Pair	$x(n),\ n \geq 0$	$X(z)$
1	$\delta(n)$	1
2	1	$\frac{z}{z-1}$
3	n	$\frac{z}{(z-1)^2}$
4	a^n	$\frac{z}{z-a}$
5	$a^n n$	$\frac{az}{(z-a)^2}$
6	$\cos(\Omega_0 n)$	$\frac{z(z-\cos\Omega_0)}{z^2-2(\cos\Omega_0)z+1}$
7	$\sin(\Omega_0 n)$	$\frac{z\sin\Omega_0}{z^2-2(\cos\Omega_0)z+1}$
8	$\lvert a\rvert^n \cos(\Omega_0 n)$	$\frac{z(z-\lvert a\rvert\cos\Omega_0)}{z^2-2\lvert a\rvert(\cos\Omega_0)z+\lvert a\rvert^2}$
9	$\lvert a\rvert^n \sin(\Omega_0 n)$	$\frac{z\lvert a\rvert\sin\Omega_0}{z^2-2\lvert a\rvert(\cos\Omega_0)z+\lvert a\rvert^2}$

Solving Difference Equations [Using the Unilateral z Transform]

- Many systems of interest in engineering applications can be characterized by constant-coefficient linear difference equations.
- One common use of the unilateral z transform is in solving constant-coefficient linear difference equations with nonzero initial conditions.

Part 13

Complex Analysis

Complex Numbers

- A **complex number** is a number of the form $z = x + jy$ where x and y are real numbers and j is the constant defined by $j^2 = -1$ (i.e., $j = \sqrt{-1}$).
- The **Cartesian form** of the complex number z expresses z in the form

$$z = x + jy,$$

where x and y are real numbers. The quantities x and y are called the **real part** and **imaginary part** of z, and are denoted as $\mathrm{Re}\, z$ and $\mathrm{Im}\, z$, respectively.

- The **polar form** of the complex number z expresses z in the form

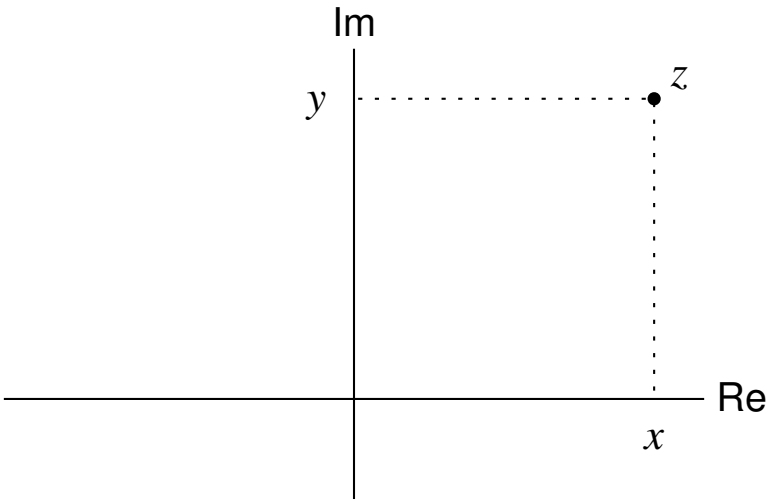
$$z = r(\cos\theta + j\sin\theta) \quad \text{or equivalently} \quad z = re^{j\theta},$$

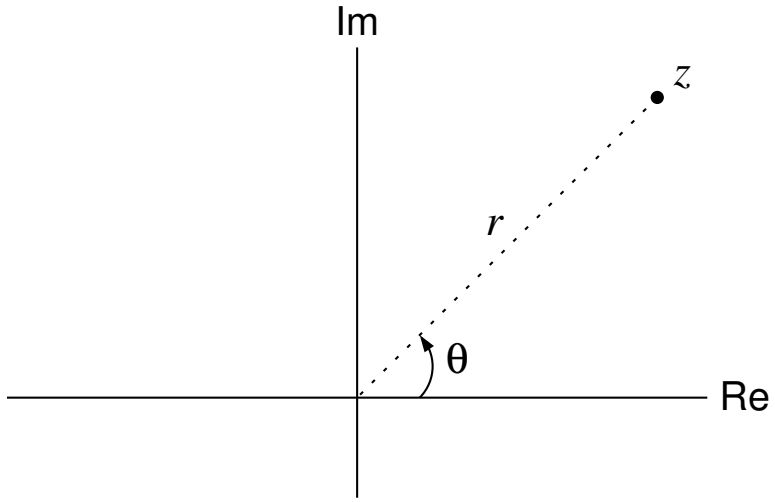
where r and θ are real numbers and $r \geq 0$. The quantities r and θ are called the **magnitude** and **argument** of z, and are denoted as $|z|$ and $\arg z$, respectively. [Note: $e^{j\theta} = \cos\theta + j\sin\theta$.]

Complex Numbers (Continued)

- Since $e^{j\theta} = e^{j(\theta+2\pi k)}$ for all real θ and all integer k, the argument of a complex number is only uniquely determined to within an additive multiple of 2π.
- The **principal argument** of a complex number z, denoted $\operatorname{Arg} z$, is the particular value θ of $\arg z$ that satisfies $-\pi < \theta \leq \pi$.
- The principal argument of a complex number (excluding zero) is ***unique***.

Geometric Interpretation of Cartesian and Polar Forms





Cartesian form:

$$z = x + jy$$

where $x = \mathrm{Re}\, z$ and $y = \mathrm{Im}\, z$

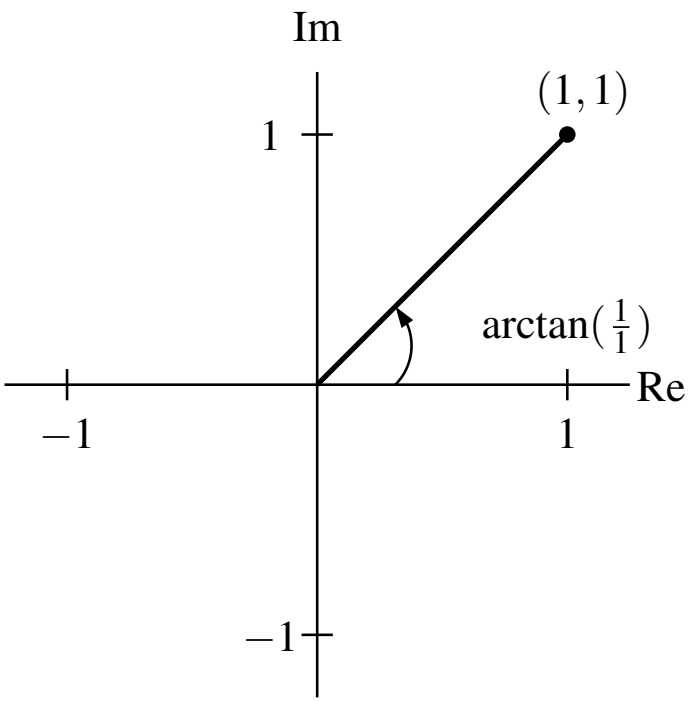
Polar form:

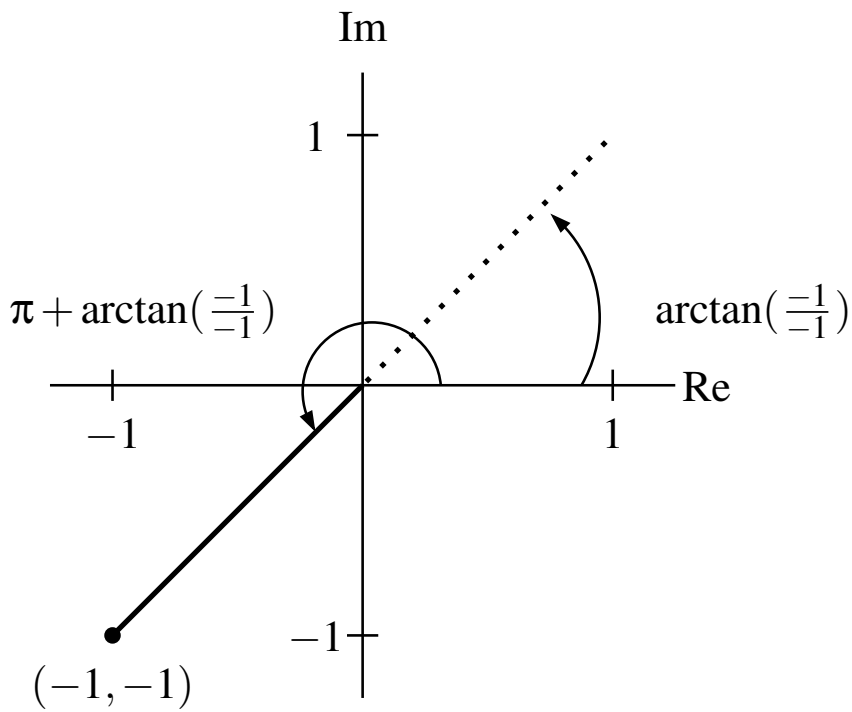
$$z = r(\cos\theta + j\sin\theta) = re^{j\theta}$$

where $r = |z|$ and $\theta = \arg z$

The arctan Function

- The range of the arctan function is $-\pi/2$ (exclusive) to $\pi/2$ (exclusive).
- Consequently, the arctan function always yields an angle in either the first or fourth quadrant.





The atan2 Function

- The angle θ that a vector from the origin to the point (x,y) makes with the positive x axis is given by $\theta = \text{atan2}(y,x)$, where

$$\text{atan2}(y,x) \triangleq \begin{cases} \arctan(y/x) & x > 0 \\ \pi/2 & x = 0 \text{ and } y > 0 \\ -\pi/2 & x = 0 \text{ and } y < 0 \\ \arctan(y/x) + \pi & x < 0 \text{ and } y \geq 0 \\ \arctan(y/x) - \pi & x < 0 \text{ and } y < 0. \end{cases}$$

- The range of the atan2 function is from $-\pi$ (exclusive) to π (inclusive).
- For the complex number z expressed in Cartesian form $x + jy$, $\text{Arg}\, z = \text{atan2}(y,x)$.
- Although the atan2 function is quite useful for computing the principal argument (or argument) of a complex number, it is ***not advisable to memorize*** the definition of this function. It is better to simply understand what this function is doing (namely, intelligently applying the arctan function).

Conversion Between Cartesian and Polar Form

- Let z be a complex number with the Cartesian and polar form representations given respectively by

$$z = x + jy \quad \text{and} \quad z = re^{j\theta}.$$

- To convert from ***polar to Cartesian*** form, we use the following identities:

$$x = r\cos\theta \quad \text{and} \quad y = r\sin\theta.$$

- To convert from ***Cartesian to polar*** form, we use the following identities:

$$r = \sqrt{x^2 + y^2} \quad \text{and} \quad \theta = \operatorname{atan2}(y,x) + 2\pi k,$$

  where k is an arbitrary integer.

- Since the $\operatorname{atan2}$ function simply amounts to the intelligent application of the $\arctan$ function, instead of memorizing the definition of the $\operatorname{atan2}$ function, one should simply ***understand*** how to use the $\arctan$ function to achieve the same result.

Properties of Complex Numbers

- For complex numbers, addition and multiplication are ***commutative***. That is, for any two complex numbers z_1 and z_2,

$$z_1 + z_2 = z_2 + z_1 \quad \text{and}$$
$$z_1 z_2 = z_2 z_1.$$

- For complex numbers, addition and multiplication are ***associative***. That is, for any three complex numbers z_1, z_2, and z_3,

$$(z_1 + z_2) + z_3 = z_1 + (z_2 + z_3) \quad \text{and}$$
$$(z_1 z_2) z_3 = z_1 (z_2 z_3).$$

- For complex numbers, the ***distributive*** property holds. That is, for any three complex numbers z_1, z_2, and z_3,

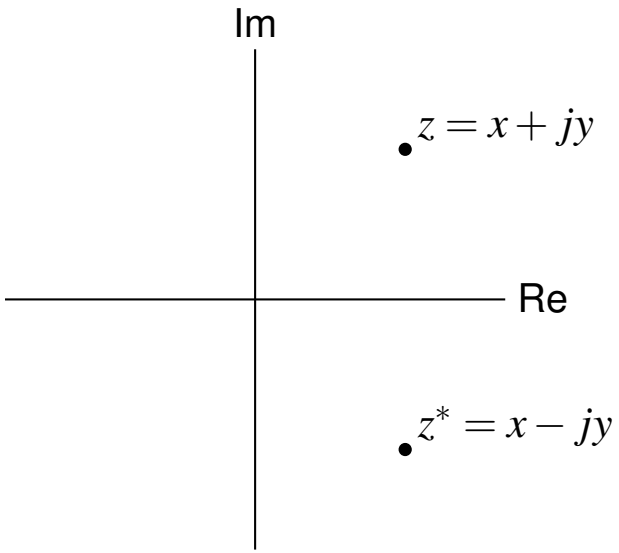
$$z_1 (z_2 + z_3) = z_1 z_2 + z_1 z_3.$$

Conjugation

- The **conjugate** of the complex number $z = x + jy$ is denoted as z^* and defined as

$$z^* = x - jy.$$

- Geometrically, the conjugation operation reflects a point in the complex plane about the real axis.
- The geometric interpretation of the conjugate is illustrated below.



Properties of Conjugation

- For every complex number z, the following identities hold:

$$|z^*| = |z|,$$
$$\arg z^* = -\arg z,$$
$$zz^* = |z|^2,$$
$$\operatorname{Re} z = \tfrac{1}{2}(z + z^*), \quad \text{and}$$
$$\operatorname{Im} z = \tfrac{1}{2j}(z - z^*).$$

- For all complex numbers z_1 and z_2, the following identities hold:

$$(z_1 + z_2)^* = z_1^* + z_2^*,$$
$$(z_1 z_2)^* = z_1^* z_2^*, \quad \text{and}$$
$$(z_1 / z_2)^* = z_1^* / z_2^*.$$

Addition

- ***Cartesian form:*** Let $z_1 = x_1 + jy_1$ and $z_2 = x_2 + jy_2$. Then,

$$\begin{aligned} z_1 + z_2 &= (x_1 + jy_1) + (x_2 + jy_2) \\ &= (x_1 + x_2) + j(y_1 + y_2). \end{aligned}$$

- That is, to add complex numbers expressed in Cartesian form, we simply add their real parts and add their imaginary parts.
- ***Polar form:*** Let $z_1 = r_1 e^{j\theta_1}$ and $z_2 = r_2 e^{j\theta_2}$. Then,

$$\begin{aligned} z_1 + z_2 &= r_1 e^{j\theta_1} + r_2 e^{j\theta_2} \\ &= (r_1 \cos\theta_1 + jr_1 \sin\theta_1) + (r_2 \cos\theta_2 + jr_2 \sin\theta_2) \\ &= (r_1 \cos\theta_1 + r_2 \cos\theta_2) + j(r_1 \sin\theta_1 + r_2 \sin\theta_2). \end{aligned}$$

- That is, to add complex numbers expressed in polar form, we first rewrite them in Cartesian form, and then add their real parts and add their imaginary parts.
- For the purposes of addition, it is easier to work with complex numbers expressed in Cartesian form.

Multiplication

- ***Cartesian form:*** Let $z_1 = x_1 + jy_1$ and $z_2 = x_2 + jy_2$. Then,

$$\begin{aligned} z_1 z_2 &= (x_1 + jy_1)(x_2 + jy_2) \\ &= x_1 x_2 + jx_1 y_2 + jx_2 y_1 - y_1 y_2 \\ &= (x_1 x_2 - y_1 y_2) + j(x_1 y_2 + x_2 y_1). \end{aligned}$$

- That is, to multiply two complex numbers expressed in Cartesian form, we use the distributive law along with the fact that $j^2 = -1$.
- ***Polar form:*** Let $z_1 = r_1 e^{j\theta_1}$ and $z_2 = r_2 e^{j\theta_2}$. Then,

$$z_1 z_2 = \left(r_1 e^{j\theta_1}\right)\left(r_2 e^{j\theta_2}\right) = r_1 r_2 e^{j(\theta_1 + \theta_2)}.$$

- That is, to multiply two complex numbers expressed in polar form, we use exponent rules.
- For the purposes of multiplication, it is easier to work with complex numbers expressed in polar form.

Division

- ***Cartesian form:*** Let $z_1 = x_1 + jy_1$ and $z_2 = x_2 + jy_2$. Then,

$$\frac{z_1}{z_2} = \frac{z_1 z_2^*}{z_2 z_2^*} = \frac{z_1 z_2^*}{|z_2|^2} = \frac{(x_1 + jy_1)(x_2 - jy_2)}{x_2^2 + y_2^2}$$
$$= \frac{x_1 x_2 - jx_1 y_2 + jx_2 y_1 + y_1 y_2}{x_2^2 + y_2^2} = \frac{x_1 x_2 + y_1 y_2 + j(x_2 y_1 - x_1 y_2)}{x_2^2 + y_2^2}.$$

- That is, to compute the quotient of two complex numbers expressed in Cartesian form, we convert the problem into one of division by a real number.
- ***Polar form:*** Let $z_1 = r_1 e^{j\theta_1}$ and $z_2 = r_2 e^{j\theta_2}$. Then,

$$\frac{z_1}{z_2} = \frac{r_1 e^{j\theta_1}}{r_2 e^{j\theta_2}} = \frac{r_1}{r_2} e^{j(\theta_1 - \theta_2)}.$$

- That is, to compute the quotient of two complex numbers expressed in polar form, we use exponent rules.
- For the purposes of division, it is easier to work with complex numbers expressed in polar form.

Properties of the Magnitude and Argument

- For any complex numbers z_1 and z_2, the following identities hold:

$$|z_1 z_2| = |z_1||z_2|,$$

$$\left|\frac{z_1}{z_2}\right| = \frac{|z_1|}{|z_2|} \quad \text{for } z_2 \neq 0,$$

$$\arg z_1 z_2 = \arg z_1 + \arg z_2, \quad \text{and}$$

$$\arg\left(\frac{z_1}{z_2}\right) = \arg z_1 - \arg z_2 \quad \text{for } z_2 \neq 0.$$

- The above properties trivially follow from the polar representation of complex numbers.

Euler's Relation and De Moivre's Theorem

- **Euler's relation.** For all real θ,

$$e^{j\theta} = \cos\theta + j\sin\theta.$$

- From Euler's relation, we can deduce the following useful identities:

$$\cos\theta = \tfrac{1}{2}(e^{j\theta} + e^{-j\theta}) \quad \text{and}$$
$$\sin\theta = \tfrac{1}{2j}(e^{j\theta} - e^{-j\theta}).$$

- **De Moivre's theorem.** For all real θ and all ***integer*** n,

$$e^{jn\theta} = \left(e^{j\theta}\right)^n.$$

[Note: This relationship does not necessarily hold for ***real*** n.]

Roots of Complex Numbers

- Every complex number $z = re^{j\theta}$ (where $r = |z|$ and $\theta = \arg z$) has n distinct ***nth roots*** given by

$$\sqrt[n]{r}e^{j(\theta+2\pi k)/n} \quad \text{for } k = 0, 1, \ldots, n-1.$$

- For example, 1 has the two distinct square roots 1 and -1.

Quadratic Formula

- Consider the equation

$$az^2 + bz + c = 0,$$

  where a, b, and c are real, z is complex, and $a \neq 0$.

- The roots of this equation are given by

$$z = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

- This formula is often useful in factoring quadratic polynomials.
- The quadratic $az^2 + bz + c$ can be factored as $a(z - z_0)(z - z_1)$, where

$$z_0 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad z_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$$

Complex Functions

- A **complex function** maps complex numbers to complex numbers. For example, the function $F(z) = z^2 + 2z + 1$, where z is complex, is a complex function.
- A complex **polynomial function** is a mapping of the form

$$F(z) = a_0 + a_1 z + a_2 z^2 + \cdots + a_n z^n,$$

  where z, $a_0, a_1, \ldots, a_n$ are complex.
- A complex **rational function** is a mapping of the form

$$F(z) = \frac{a_0 + a_1 z + a_2 z^2 + \ldots + a_n z^n}{b_0 + b_1 z + b_2 z^2 + \ldots + b_m z^m},$$

  where $a_0, a_1, \ldots, a_n, b_0, b_1, \ldots, b_m$ and z are complex.
- Observe that a polynomial function is a special case of a rational function.
- Herein, we will mostly focus our attention on polynomial and rational functions.

Continuity

- A function F is said to be **continuous at a point** z_0 if $F(z_0)$ is defined and given by

$$F(z_0) = \lim_{z \to z_0} F(z).$$

- A function that is continuous at every point in its domain is said to be **continuous**.
- Polynomial functions are continuous everywhere.
- Rational functions are continuous everywhere except at points where the denominator polynomial becomes zero.

Differentiability

- A function F is said to be **differentiable at a point** $z = z_0$ if the limit

$$F'(z_0) = \lim_{z \to z_0} \frac{F(z) - F(z_0)}{z - z_0}$$

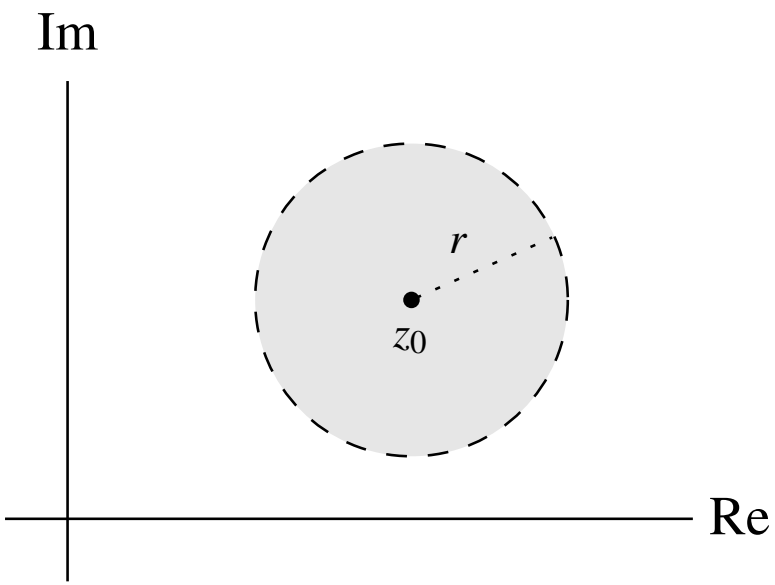
  exists. This limit is called the **derivative** of F at the point $z = z_0$.
- A function is said to be **differentiable** if it is differentiable at every point in its domain.
- The rules for differentiating sums, products, and quotients are the same for complex functions as for real functions. If $F'(z_0)$ and $G'(z_0)$ exist, then
  1. $(aF)'(z_0) = aF'(z_0)$ for any complex constant a;
  2. $(F+G)'(z_0) = F'(z_0) + G'(z_0)$;
  3. $(FG)'(z_0) = F'(z_0)G(z_0) + F(z_0)G'(z_0)$;
  4. $(F/G)'(z_0) = \frac{G(z_0)F'(z_0) - F(z_0)G'(z_0)}{G(z_0)^2}$; and
  5. if $z_0 = G(w_0)$ and $G'(w_0)$ exists, then the derivative of $F(G(z))$ at w_0 is $F'(z_0)G'(w_0)$ (i.e., the chain rule).
- A polynomial function is differentiable everywhere.
- A rational function is differentiable everywhere except at the points where its denominator polynomial becomes zero.

Open Disks

- An **open disk** in the complex plane with center z_0 and radius r is the set of complex numbers z satisfying

$$|z - z_0| < r,$$

  where r is a strictly positive real number.
- A plot of an open disk is shown below.



Analyticity

- A function is said to be **analytic at a point** z_0 if it is differentiable at every point in an open disk about z_0.
- A function is said to be **analytic** if it is analytic at every point in its domain.
- A polynomial function is analytic everywhere.
- A rational function is analytic everywhere, except at the points where its denominator polynomial becomes zero.

Zeros and Singularities

- If a function F is zero at the point z_0 (i.e., $F(z_0) = 0$), F is said to have a **zero** at z_0.
- If a function F is such that $F(z_0) = 0, F^{(1)}(z_0) = 0, \ldots, F^{(n-1)}(z_0) = 0$ (where $F^{(k)}$ denotes the kth order derivative of F), F is said to have an **nth order zero** at z_0.
- A point at which a function fails to be analytic is called a **singularity**.
- Polynomials do not have singularities.
- Rational functions can have a type of singularity called a pole.
- If a function F is such that $G(z) = 1/F(z)$ has an nth order zero at z_0, F is said to have an **nth order pole** at z_0.
- A pole of first order is said to be **simple**, whereas a pole of order two or greater is said to be **repeated**. A similar terminology can also be applied to zeros (i.e., **simple zero** and **repeated zero**).

Zeros and Poles of a Rational Function

- Given a rational function F, we can always express F in factored form as

$$F(z) = \frac{K(z-a_1)^{\alpha_1}(z-a_2)^{\alpha_2}\cdots(z-a_M)^{\alpha_M}}{(z-b_1)^{\beta_1}(z-b_2)^{\beta_2}\cdots(z-b_N)^{\beta_N}},$$

  where K is complex, $a_1, a_2, \ldots, a_M, b_1, b_2, \ldots, b_N$ are distinct complex numbers, and $\alpha_1, \alpha_2, \ldots, \alpha_M$ and $\beta_1, \beta_2, \ldots, \beta_N$ are strictly positive integers.
- One can show that F has ***poles*** at $b_1, b_2, \ldots, b_N$ and ***zeros*** at $a_1, a_2, \ldots, a_M$.
- Furthermore, the kth pole (i.e., b_k) is of ***order*** β_k, and the kth zero (i.e., a_k) is of ***order*** α_k.
- When plotting zeros and poles in the complex plane, the symbols “o” and “x” are used to denote zeros and poles, respectively.

Part 14

Partial Fraction Expansions (PFEs)

Motivation for PFEs

- Sometimes it is beneficial to be able to express a rational function as a sum of ***lower-order*** rational functions.
- This can be accomplished using a type of decomposition known as a partial fraction expansion.
- Partial fraction expansions are often useful in the calculation of inverse Laplace transforms, inverse z transforms, and inverse CT/DT Fourier transforms.

Strictly-Proper Rational Functions

- Consider a rational function

$$F(v) = \frac{\alpha_m v^m + \alpha_{m-1} v^{m-1} + \ldots + \alpha_1 v + \alpha_0}{\beta_n v^n + \beta_{n-1} v^{n-1} + \ldots + \beta_1 v + \beta_0}.$$

- The function F is said to be **strictly proper** if $m < n$ (i.e., the order of the numerator polynomial is strictly less than the order of the denominator polynomial).
- Through polynomial long division, any rational function can be written as the sum of a polynomial and a strictly-proper rational function.
- A ***strictly-proper*** rational function can be expressed as a sum of lower-order rational functions, with such an expression being called a partial fraction expansion.

Section 14.1

PFEs for First Form of Rational Functions

Partial Fraction Expansions (PFEs) [CT and DT Contexts]

- Any rational function F can be expressed in the form of

$$F(v) = \frac{a_m v^m + a_{m-1} v^{m-1} + \ldots + a_0}{v^n + b_{n-1} v^{n-1} + \ldots + b_0}.$$

- Furthermore, the denominator polynomial $D(v) = v^n + b_{n-1} v^{n-1} + \ldots + b_0$ in the above expression for $F(v)$ can be factored to obtain

$$D(v) = (v - p_1)^{q_1} (v - p_2)^{q_2} \cdots (v - p_n)^{q_n},$$

  where the p_k are distinct and the q_k are integers.
- If F has only simple poles, $q_1 = q_2 = \cdots = q_n = 1$.
- Suppose that F is strictly proper (i.e., $m < n$).
- In the determination of a partial fraction expansion of F, there are ***two cases*** to consider:
  1. F has ***only simple poles***; and
  2. F has ***at least one repeated pole***.

- Suppose that the (rational) function F has only simple poles.
- Then, the denominator polynomial D for F is of the form

$$D(v) = (v - p_1)(v - p_2) \cdots (v - p_n),$$

where the p_k are distinct.

- In this case, F has a partial fraction expansion of the form

$$F(v) = \frac{A_1}{v - p_1} + \frac{A_2}{v - p_2} + \ldots + \frac{A_{n-1}}{v - p_{n-1}} + \frac{A_n}{v - p_n},$$

where

$$A_k = (v - p_k)F(v)\big|_{v=p_k}.$$

- Note that the (simple) pole p_k contributes a single term to the partial fraction expansion.

Repeated-Pole Case [CT and DT Contexts]

- Suppose that the (rational) function F has at least one repeated pole.
- In this case, F has a partial fraction expansion of the form

$$F(v) = \left[\frac{A_{1,1}}{v-p_1} + \frac{A_{1,2}}{(v-p_1)^2} + \ldots + \frac{A_{1,q_1}}{(v-p_1)^{q_1}}\right] + \left[\frac{A_{2,1}}{v-p_2} + \ldots + \frac{A_{2,q_2}}{(v-p_2)^{q_2}}\right] + \ldots + \left[\frac{A_{P,1}}{v-p_P} + \ldots + \frac{A_{P,q_P}}{(v-p_P)^{q_P}}\right],$$

  where

$$A_{k,\ell} = \frac{1}{(q_k-\ell)!}\left[\left[\frac{d}{dv}\right]^{q_k-\ell}\left[(v-p_k)^{q_k}F(v)\right]\right]\Big|_{v=p_k}.$$

- Note that the q_kth-order pole p_k contributes q_k terms to the partial fraction expansion.
- Note that $n! = (n)(n-1)(n-2)\cdots(1)$ and $0! = 1$.

Section 14.2

PFEs for Second Form of Rational Functions

Partial Fraction Expansions (PFEs) [DT Context]

- Any rational function F can be expressed in the form of

$$F(v) = \frac{a_m v^m + a_{m-1} v^{m-1} + \ldots + a_1 v + a_0}{b_n v^n + b_{n-1} v^{n-1} + \ldots + b_1 v + 1}.$$

- Furthermore, the denominator polynomial $D(v) = b_n v^n + b_{n-1} v^{n-1} + \ldots + b_1 v + 1$ in the above expression for $F(v)$ can be factored to obtain

$$D(v) = (1 - p_1^{-1} v)^{q_1} (1 - p_2^{-1} v)^{q_2} \cdots (1 - p_n^{-1} v)^{q_n},$$

  where the p_k are distinct and the q_k are integers.

- If F has only simple poles, $q_1 = q_2 = \cdots = q_n = 1$.
- Suppose that F is strictly proper (i.e., $m < n$).
- In the determination of a partial fraction expansion of F, there are ***two cases*** to consider:
  1. F has ***only simple poles***; and
  2. F has ***at least one repeated pole***.

Simple-Pole Case [DT Context]

- Suppose that the (rational) function F has only simple poles.
- Then, the denominator polynomial D for F is of the form

$$D(v) = (1 - p_1^{-1}v)(1 - p_2^{-1}v) \cdots (1 - p_n^{-1}v),$$

  where the p_k are distinct.
- In this case, F has a partial fraction expansion of the form

$$F(v) = \frac{A_1}{1 - p_1^{-1}v} + \frac{A_2}{1 - p_2^{-1}v} + \ldots + \frac{A_{n-1}}{1 - p_{n-1}^{-1}v} + \frac{A_n}{1 - p_n^{-1}v},$$

  where

$$A_k = (1 - p_k^{-1}v)F(v)\Big|_{v=p_k}.$$

- Note that the (simple) pole p_k contributes a single term to the partial fraction expansion.

Repeated-Pole Case [DT Context]

- Suppose that the (rational) function F has at least one repeated pole.
- In this case, F has a partial fraction expansion of the form

$$F(v) = \left[\frac{A_{1,1}}{1-p_1^{-1}v} + \frac{A_{1,2}}{(1-p_1^{-1}v)^2} + \ldots + \frac{A_{1,q_1}}{(1-p_1^{-1}v)^{q_1}}\right] + \left[\frac{A_{2,1}}{1-p_2^{-1}v} + \ldots + \frac{A_{2,q_2}}{(1-p_2^{-1}v)^{q_2}}\right] + \ldots + \left[\frac{A_{P,1}}{1-p_P^{-1}v} + \ldots + \frac{A_{P,q_P}}{(1-p_P^{-1}v)^{q_P}}\right],$$

  where

$$A_{k,\ell} = \frac{1}{(q_k-\ell)!}(-p_k)^{q_k-\ell}\left[\left[\frac{d}{dv}\right]^{q_k-\ell}\left[(1-p_k^{-1}v)^{q_k}F(v)\right]\right]\Big|_{v=p_k}.$$

- Note that the q_kth-order pole p_k contributes q_k terms to the partial fraction expansion.
- Note that $n! = (n)(n-1)(n-2)\cdots(1)$ and $0! = 1$.

Part 15

MATLAB

MATLAB

- MATLAB is a proprietary programming language and numeric computing environment developed by MathWorks.
- MATLAB allows such things as: matrix computation, data visualization (e.g., 2-D and 3-D plots), system simulation, implementation of algorithms, and interfacing with code written in other programming languages.
- The base functionality of MATLAB can be extended through the use of numerous toolkits.
- MATLAB is available on many platforms, including Windows, MacOS, Linux, and numerous other Unix variants.
- Extensive information on MATLAB (including detailed guides and manuals) is available from MathWorks web site:
  - `https://www.mathworks.com`.
- Relative to general-purpose programming languages such as C++ and C, MATLAB is much easier to learn, and therefore is particular well suited for teaching purposes.

- The earliest version of MATLAB was a simple interactive matrix calculator (without an associated programming language) developed by Cleve Moler (then a professor at the University of New Mexico and later a cofounder of MathWorks).
- MATLAB stands for “matrix laboratory”, with this name reflecting the roots of the software as a simple interactive matrix calculator.

GNU Octave

- GNU Octave (or simply Octave) is a software package that is very similar in functionality to MATLAB.
- In fact, Octave is largely (but not fully) compatible with MATLAB.
- Unlike MATLAB, Octave is free software and is available under the GNU General Public License.
- The Octave software was created by John W. Eaton, and was named after one of his professors, Octave Levenspiel.
- The Octave software is available for many platforms, including Windows, MacOS, Linux, and other Unix variants, and can be obtained from its official web site:
  - https://www.octave.org
- For anyone who does not have a license to use MATLAB, the Octave software provides an excellent alternative.

Section 15.1

Running MATLAB

MATLAB Command-Line Interface

- MATLAB can be invoked using a command of the following form, where *options* denotes a list of zero or more options:

  `matlab` *options*

- Some of the supported options are listed in the table below.

Option	Description
`-help`	Print help information and exit.
`-desktop`	Enable the starting of the MATLAB desktop.
`-nodesktop`	Disable the starting of the MATLAB desktop and use the terminal window for commands.
`-display` *display*	Set the X11 display to *display*.
`-nodisplay`	Do not use any X11 display.
`-nosplash`	Do not show the splash screen during startup.
`-noFigureWindows`	Disable the display of figure windows.
`-batch` *command*	Execute MATLAB command *command* upon startup without the MATLAB desktop.

- For example, to obtain help on the command-line interface of MATLAB, use the command:

  ```
  matlab -help
  ```

Remote Graphics Display Using X11

- On Unix-based systems, the graphics output of MATLAB can typically be displayed on a different machine from the one running the MATLAB program (via the X11 protocol).
- This is typically used in a scenario where the user logs into remote machine and runs the MATLAB program on this remote machine with the graphics output being directed to the user's local machine.
- In order to take advantage of this functionality, the following software is required:
  - a Secure Shell (SSH) client (that supports X11 forwarding); and
  - an X11 server.
- With the necessary configuration in place, one can simply login via SSH to another machine with MATLAB installed and then run MATLAB.
- Any graphics windows opened by the MATLAB program will appear on the local machine via its X11 server.

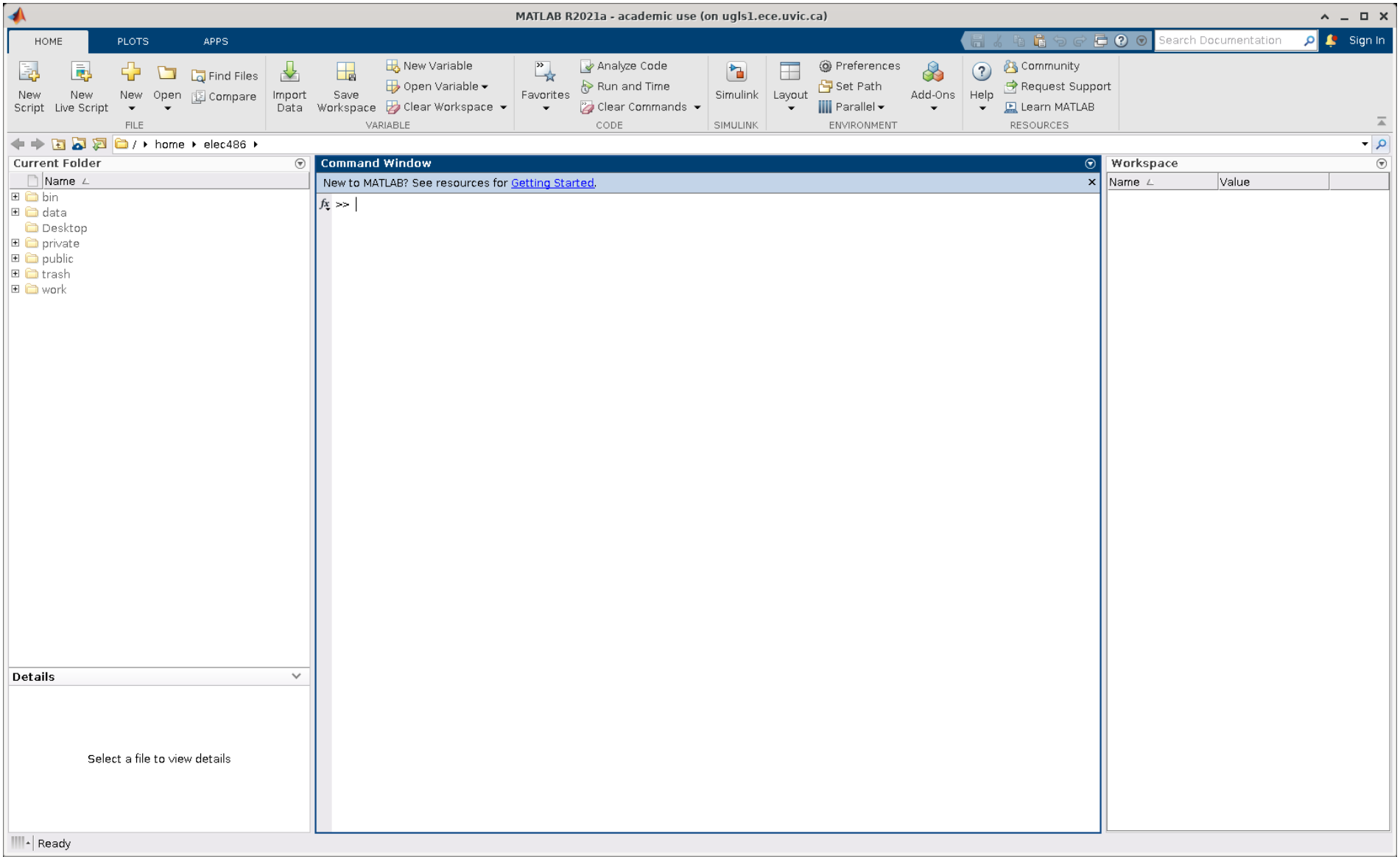
Secure Shell (SSH) Client Software

- Most Unix variants, including Linux and MacOS, typically provide SSH client software (through a program that is usually called `ssh`).
- Since recent versions of ChromeOS can run Linux in a container, the preceding comment also applies to ChromeOS. For information on containerized Linux in ChromeOS, see:
  - https://chromeos.dev/en/linux
- Some popular SSH client software for non-Unix-based platforms is as follows:
  - PuTTY
    - SSH client for Windows (and numerous Unix platforms as well)
    - open source implementation of SSH and telnet
    - web site: https://www.chiark.greenend.org.uk/~sgtatham/putty
  - MobaXterm
    - for Windows platform
    - provides both an X11 server and a SSH client
    - see slide on X11 server software for more details

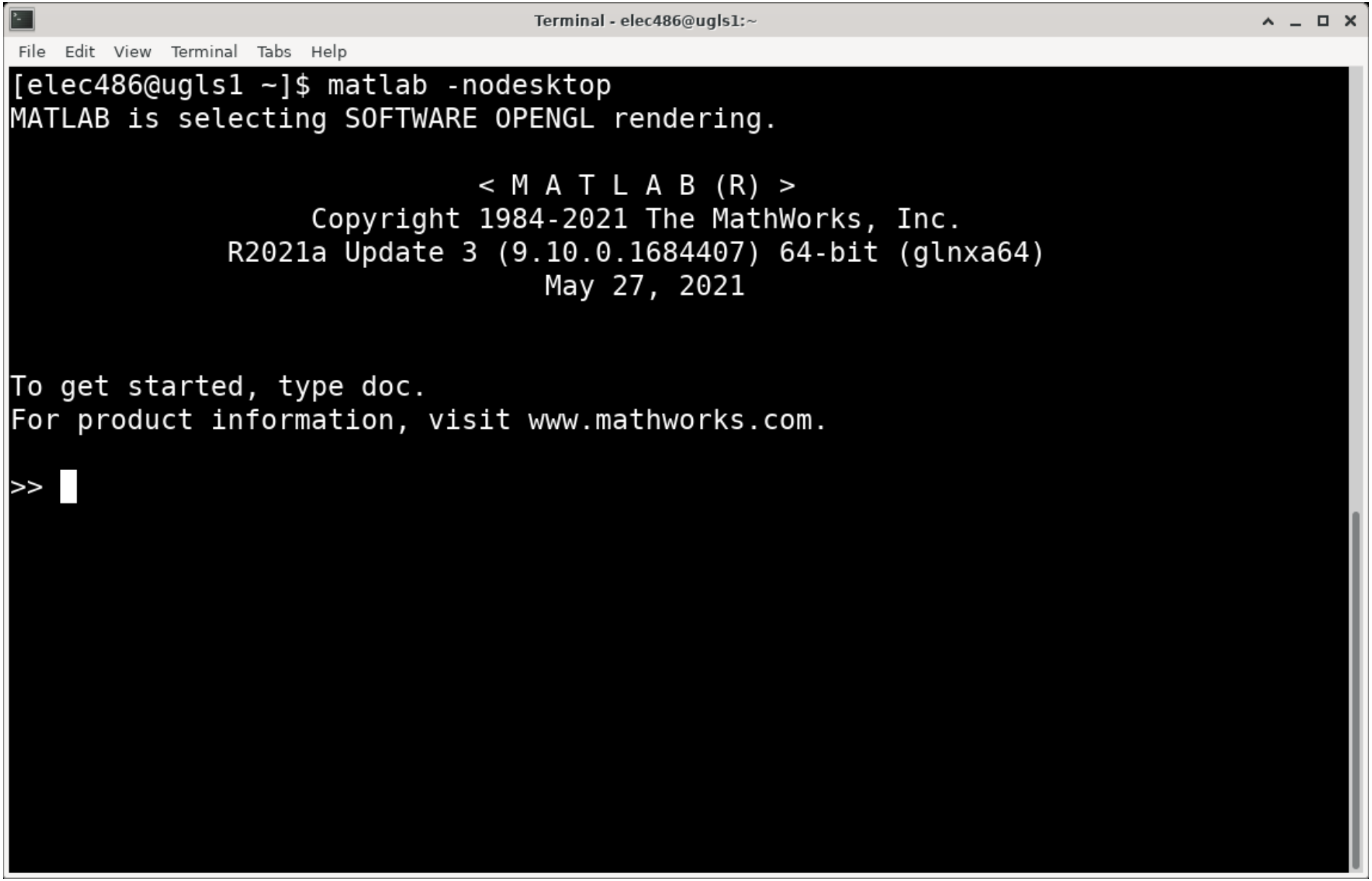
X11 Server Software

- Most Unix variants, including Linux but excluding MacOS, typically provide X11 server software.
- Some popular X11 server software for various platforms is as follows:
  - MobaXterm
    - X11 server for Windows platform
    - enhanced terminal with X11 server and SSH client
    - commercial software but Home Edition is free
    - web site: https://mobaxterm.mobatek.net
  - Xming
    - X11 server for Windows platform
    - open source
    - web site: http://www.straightrunning.com/XmingNotes
  - XQuartz
    - X11 server for MacOS platform
    - web site: https://www.xquartz.org
- For ChromeOS, use containerized Linux for an X11 server. See:
  - https://chromeos.dev/en/linux

MATLAB With Desktop



MATLAB Without Desktop

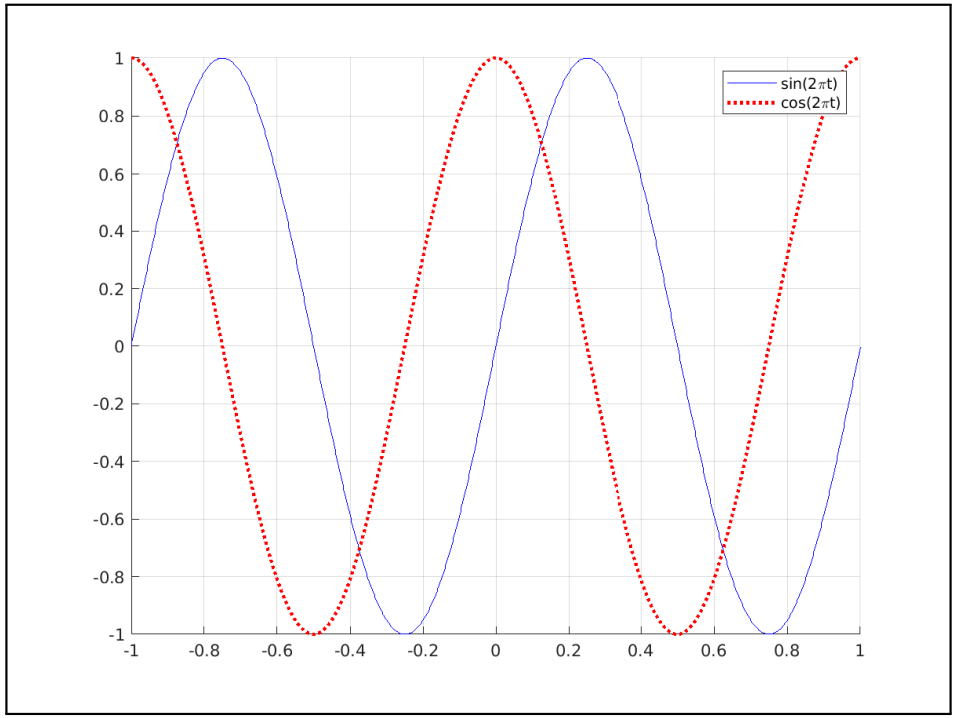


Section 15.2

MATLAB Examples

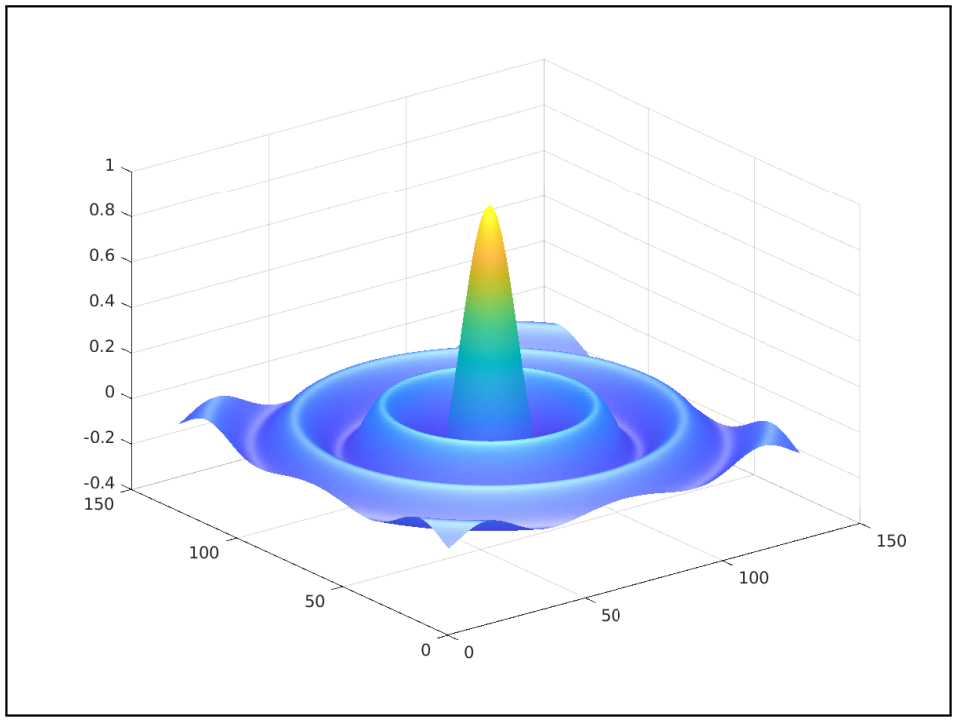
Plot of Univariate Functions

```
figure
t = linspace(-1, 1, 1024);
hold on
plot(t, sin(2 * pi * t), 'b');
plot(t, cos(2 * pi * t), 'r:', 'LineWidth', 2);
hold off
grid
legend('sin(2\pi{}t)', 'cos(2\pi{}t)');
print('output.pdf', '-dpdf', '-bestfit', '-opengl');
```



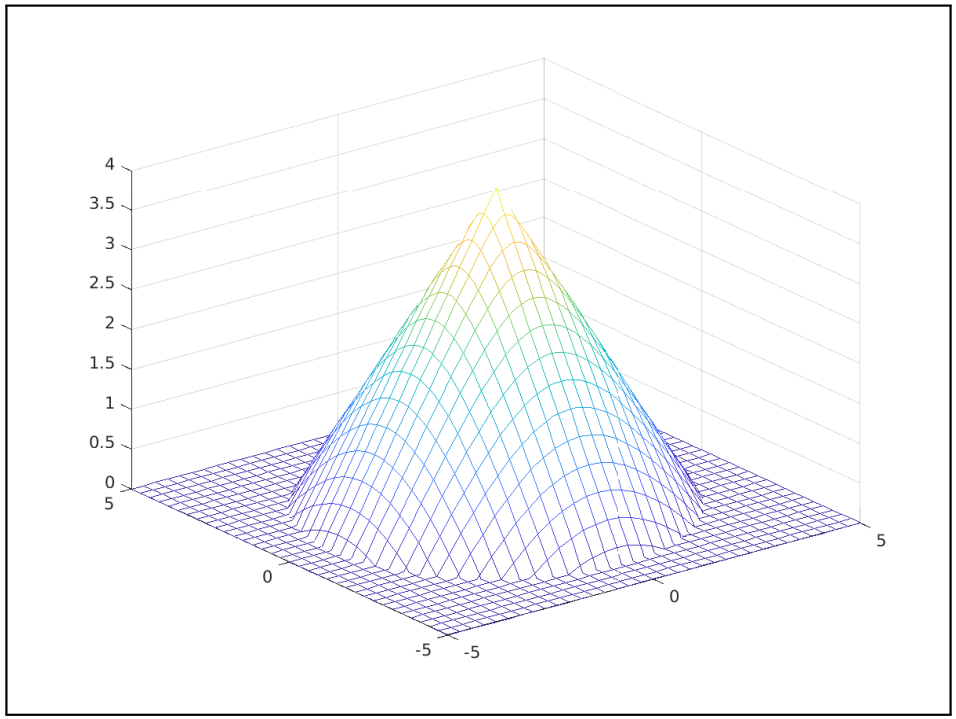
Plot of Bivariate Function With Shading

```
figure
[x y] = meshgrid(linspace(-8, 8, 129));
r = sqrt(x .^ 2 + y .^ 2);
z = sinc(2 / pi * r);
surf(z);
shading interp;
light;
lighting phong;
material([0.9 0.5 0.5 1.0]);
colormap(parula);
print('output.pdf', '-dpdf', '-bestfit');
```



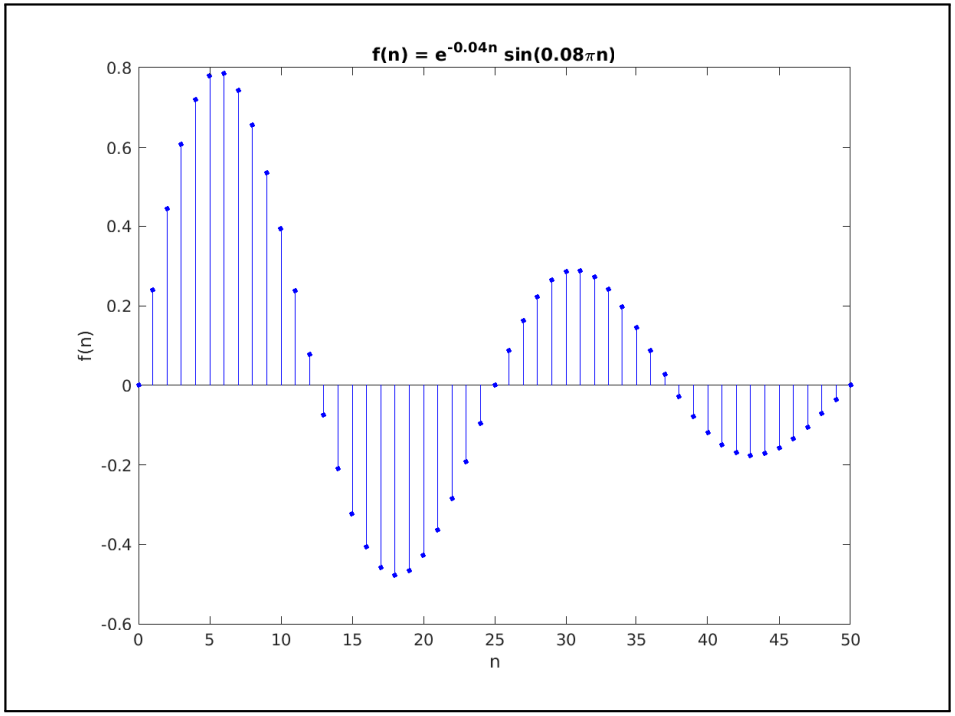
Plot of Grid-Sampled Function With Wireframe

```
1  figure
2  d = @(x,y) sqrt(x .^ 2 + y .^2);
3  f = @(x,y) (d(x,y) <= 4) .* (4 - d(x,y));
4  fmesh(f, [-5 5 -5 5]);
5  print('output.pdf', '-dpdf', '-bestfit');
```



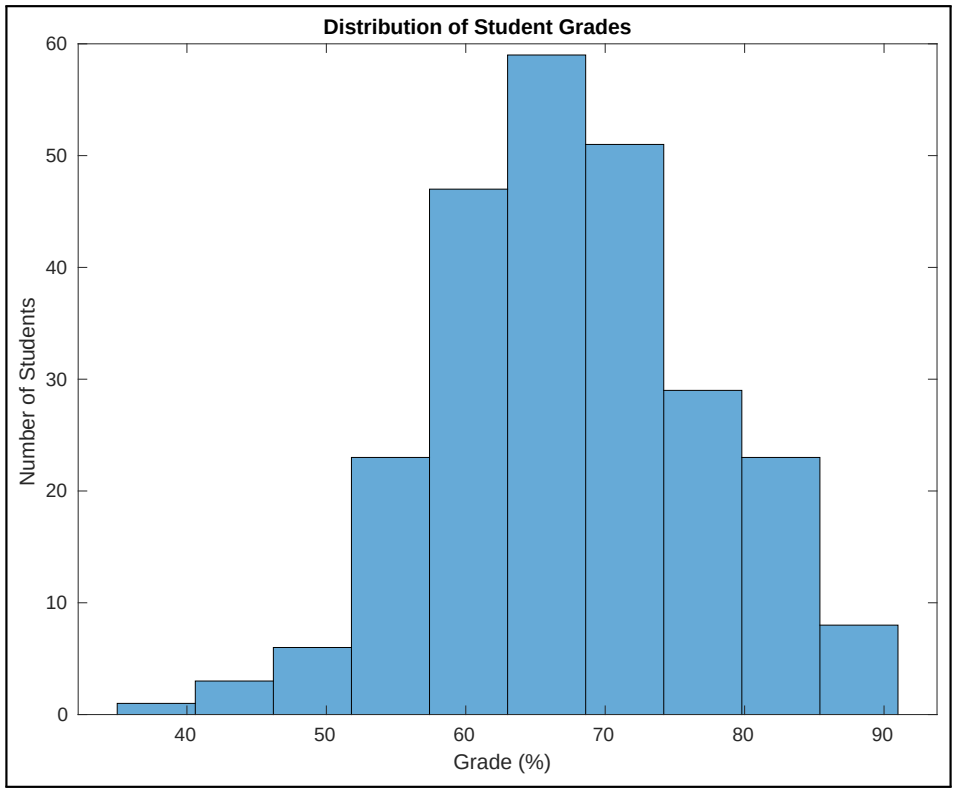
Plot of Sequence

```
figure
n = 0 : 50;
h = stem(n, exp(-0.04 * n) .* sin(0.08 * pi * n), 'filled', 'b');
set(h(1), 'MarkerSize', 3)
xlabel('n');
ylabel('f(n)');
title('f(n) = e^{-0.04n} sin(0.08{\pi}n)');
print('output.pdf', '-dpdf', '-bestfit', '-opengl');
```



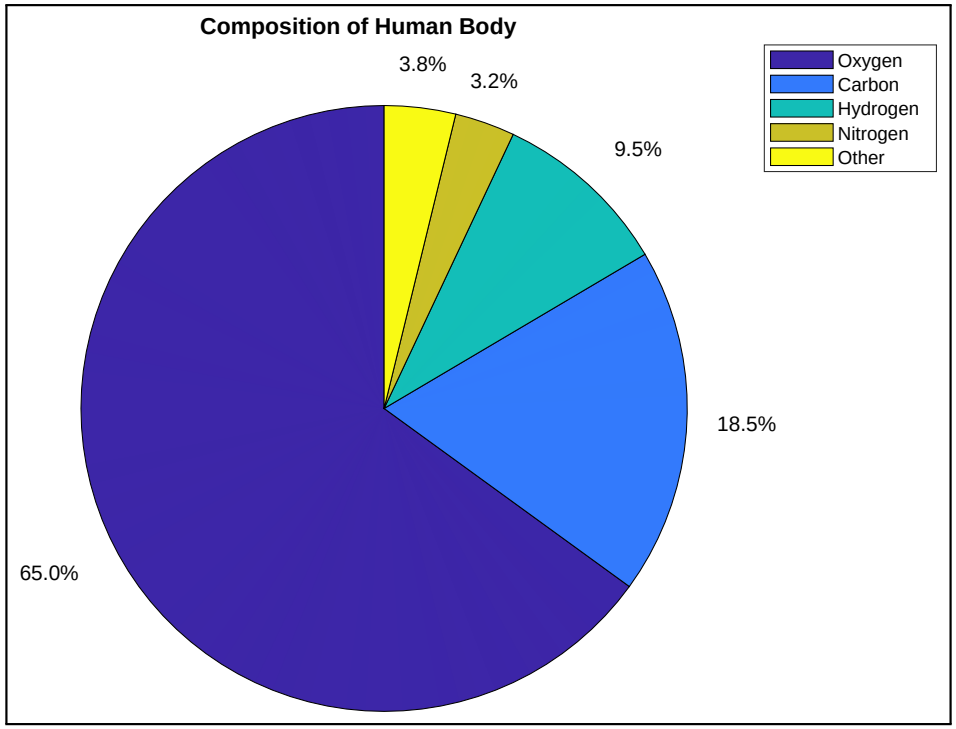
Histogram

```
clip = @(x, low, high) max(min(x, high), low);
figure
randn('seed', 5);
x = clip(67 + 10 * randn(250, 1), 0, 100);
histogram(x, 10);
title('Distribution of Student Grades');
xlabel('Grade (%)');
ylabel('Number of Students');
print('output.pdf', '-dpdf', '-bestfit');
```



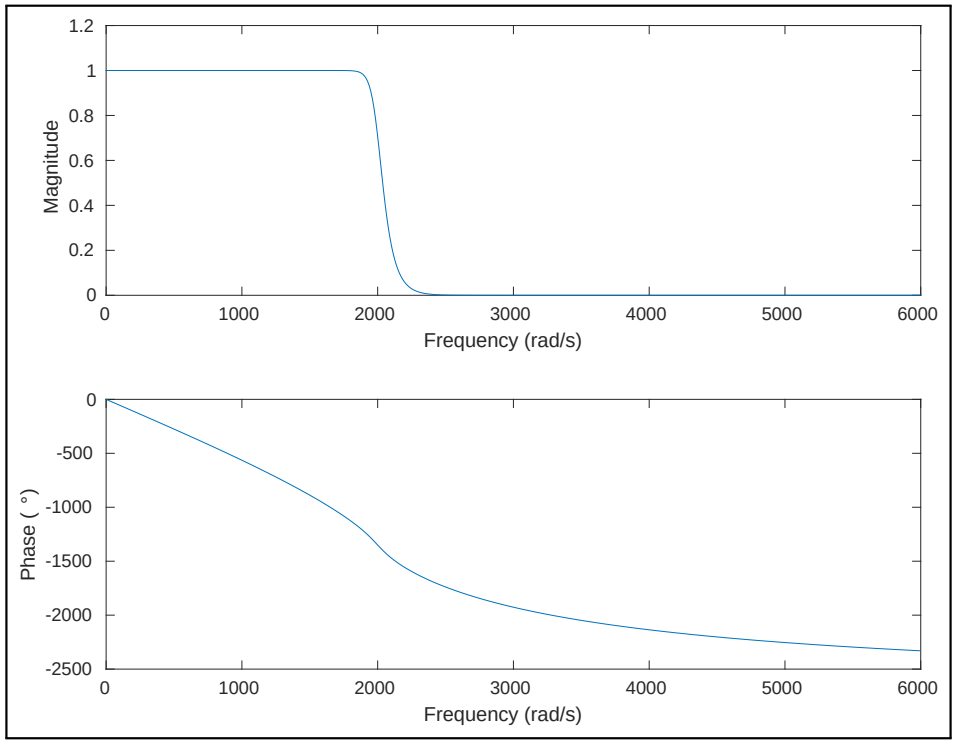
Pie Chart

```
figure
values = [65 18.5 9.5 3.2];
values = [values (100 - sum(values))];
names = {'Oxygen', 'Carbon', 'Hydrogen', 'Nitrogen', 'Other'};
explode = [0 0 0 0 0];
labels = cellstr([num2str(values', '%.1f%%' )])';
pie(values, explode, labels);
legend(names, 'Location', 'bestoutside');
title('Composition of Human Body')
print('output.pdf', '-dpdf', '-bestfit');
```



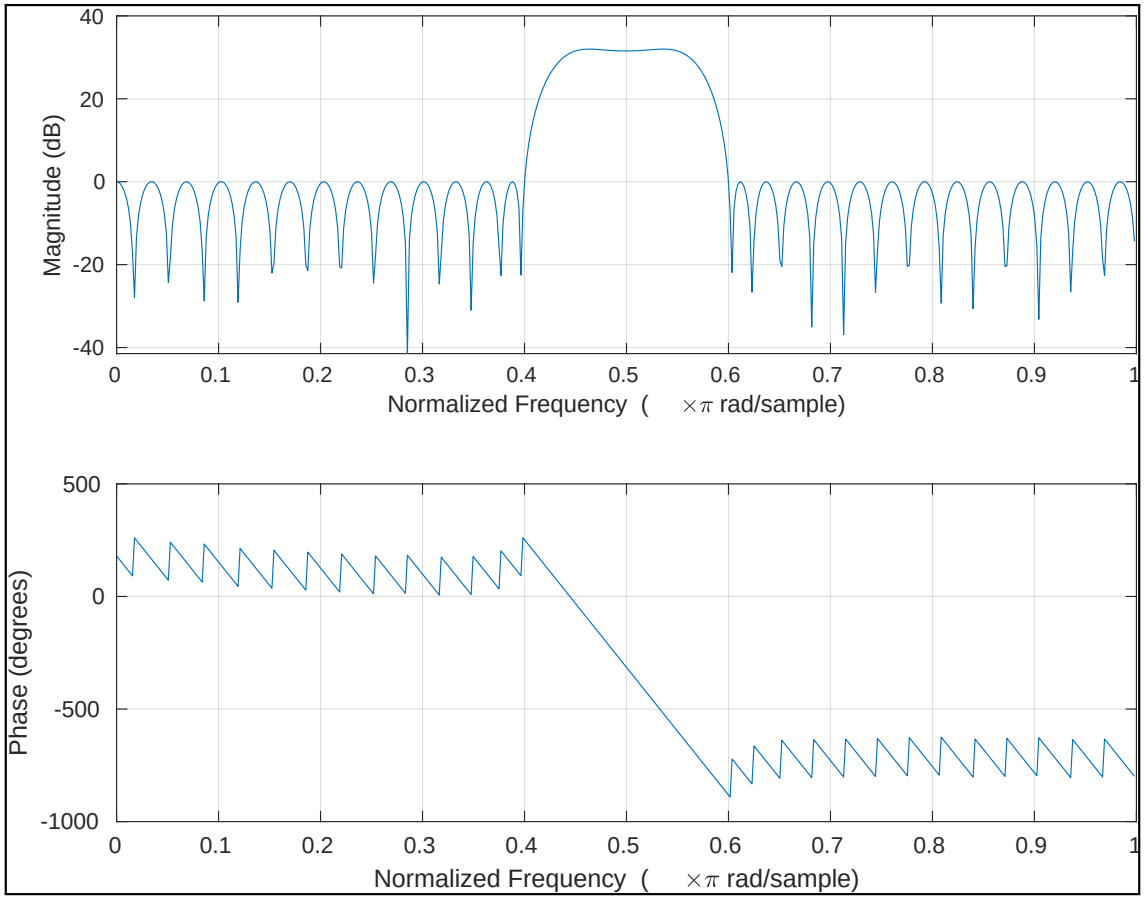
Design of Analog Lowpass Filter

```
1   figure
2   [a b] = butter(30, 2000, 's');
3   [f, w] = freqs(a, b, linspace(0, 6000, 512));
4   subplot(2, 1, 1);
5   plot(w, abs(f));
6   xlabel('Frequency (rad/s)'); ylabel('Magnitude');
7   subplot(2, 1, 2);
8   plot(w, 180 / pi * unwrap(angle(f)));
9   xlabel('Frequency (rad/s)'); ylabel('Phase (\circ)');
10  print('output.pdf', '-dpdf', '-bestfit');
```



Design of Digital Bandpass Filter

```
1  [a b] = firpm(63, [0 0.40 0.45 0.55 .60 1], [0 0 1 1 0 0]);
2  freqz(a, b);
3  print('output.pdf', '-dpdf', '-bestfit');
```



Solve Differential Equation: $y'' = 9y, y(0) = a, y'(0) = b$

```
1  syms y(t) a b
2  eqn = diff(y, t, 2) == 9 * y;
3  Dy = diff(y, t);
4  ics = [y(0) == a, Dy(0) == b];
5  ySol(t) = dsolve(eqn, ics);
6  s = evalc('pretty(ySol(t))');
7  disp(sprintf('y(t) = \n%s', s))
```

```
y(t) =
                 /  a    b \                  /  a    b \
exp(-3 t)        |  -  - -  | + exp(3 t)       |  -  + -  |
                 \  2    6 /                  \  2    6 /
```

Solve Nonlinear System: $x^2 + y^2 = a,\ 2x - y = 3$

```
1  syms x y a
2  [solx soly] = solve(x ^ 2 + y ^ 2 == a, 2 * x - y == 3, [x y]);
3  pretty([solx, soly]);
```

```
   /          / 16 a    144 \          / 16 a    144 \      \
   |      sqrt| ---- - --- |      sqrt| ---- - --- |      |
   | 6        \   5      25 /        \   5      25 /    3 |
   | - - ------------------, - ------------------ - - |
   | 5             4                   2            5 |
   |                                                    |
   |      / 16 a    144 \            / 16 a    144 \    |
   | sqrt| ---- - --- |       sqrt| ---- - --- |     |
   |      \   5      25 /    6       \   5      25 /    3 |
   | ------------------ + -,  ------------------ - - |
   \           4            5           2            5  /
```

Image Processing

```
figure
color = imread('peppers.png');
mono = rgb2gray(color);
edges = 1 - edge(mono, 'canny');
a = 8; blocky = imresize(imresize(color, 1/a), a, 'box');
subplot(2, 2, 1); set(gca, 'Position', [0 0.5 0.5 0.5])
imshow(color);
subplot(2, 2, 2); set(gca, 'Position', [0.5 0.5 0.5 0.5])
imshow(mono);
subplot(2, 2, 3); set(gca, 'Position', [0 0 0.5 0.5])
imshow(edges);
subplot(2, 2, 4); set(gca, 'Position', [0.5 0 0.5 0.5])
imshow(blocky);
print('output.pdf', '-dpdf', '-bestfit');
```

Some Basic MATLAB Commands

- The `quit` or `exit` command terminate the MATLAB program.
- The `clc` command clears the command window.
- The `clear` command clears all variables.

Section 15.3

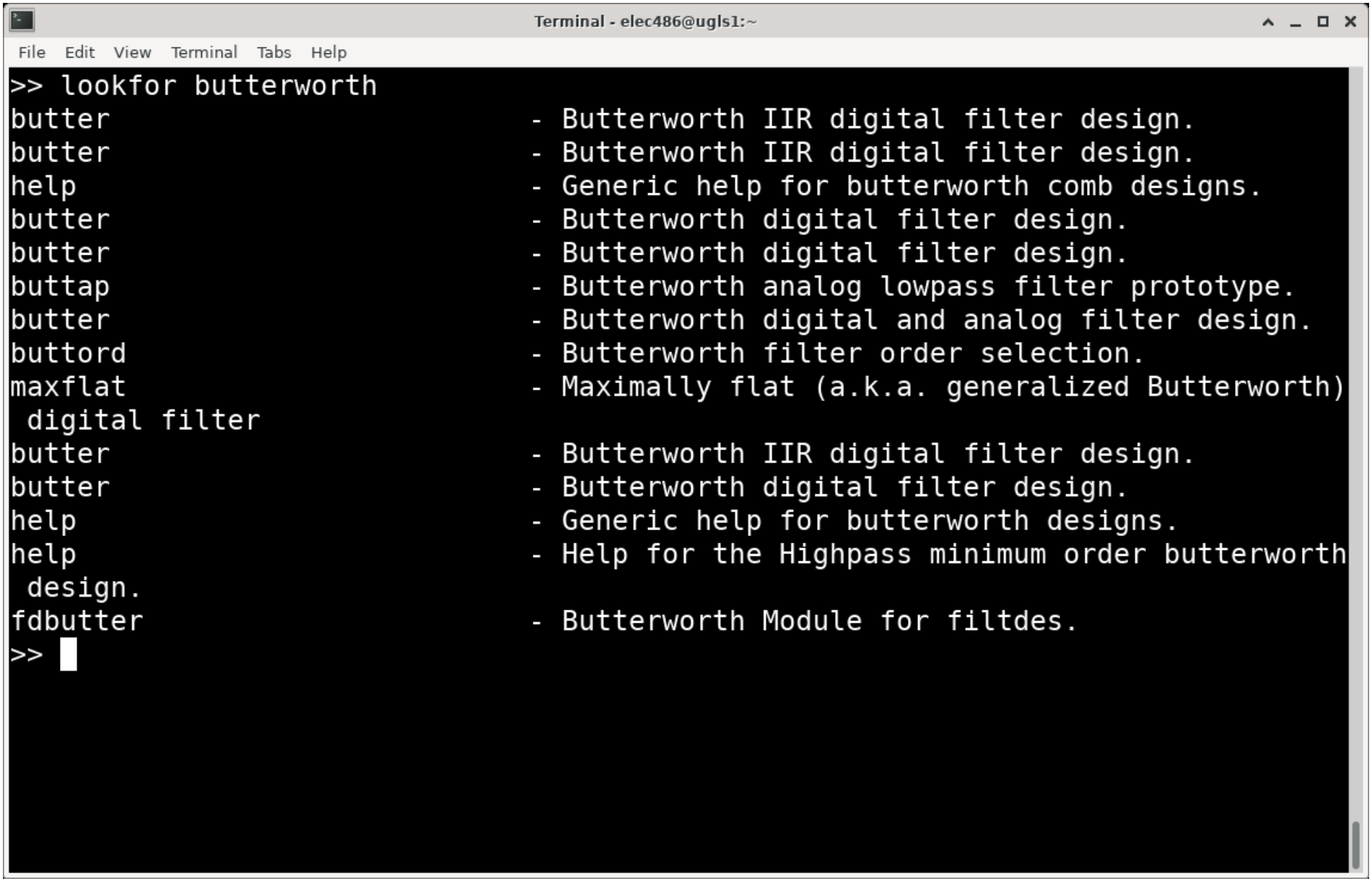
Accessing MATLAB Documentation

Searching MATLAB Help Text

- The `lookfor` command searches for a string in the first comment line of the help text in all MATLAB files found on the MATLAB search path and prints each first comment line for which a match occurs.
- For example, to search for information that contains "butterworth" (as in Butterworth filter), use the following command:

```
lookfor butterworth
```

lookfor butterworth Example



Displaying MATLAB Documentation

- Two very helpful commands for accessing documentation about various aspects of MATLAB are the `help` and `doc` commands.
- The `help` command prints information to the command window about a specified item, such as a function, command, operator symbol, or toolbox.
- For example, the following command prints information about the `abs` function:

```
help abs
```

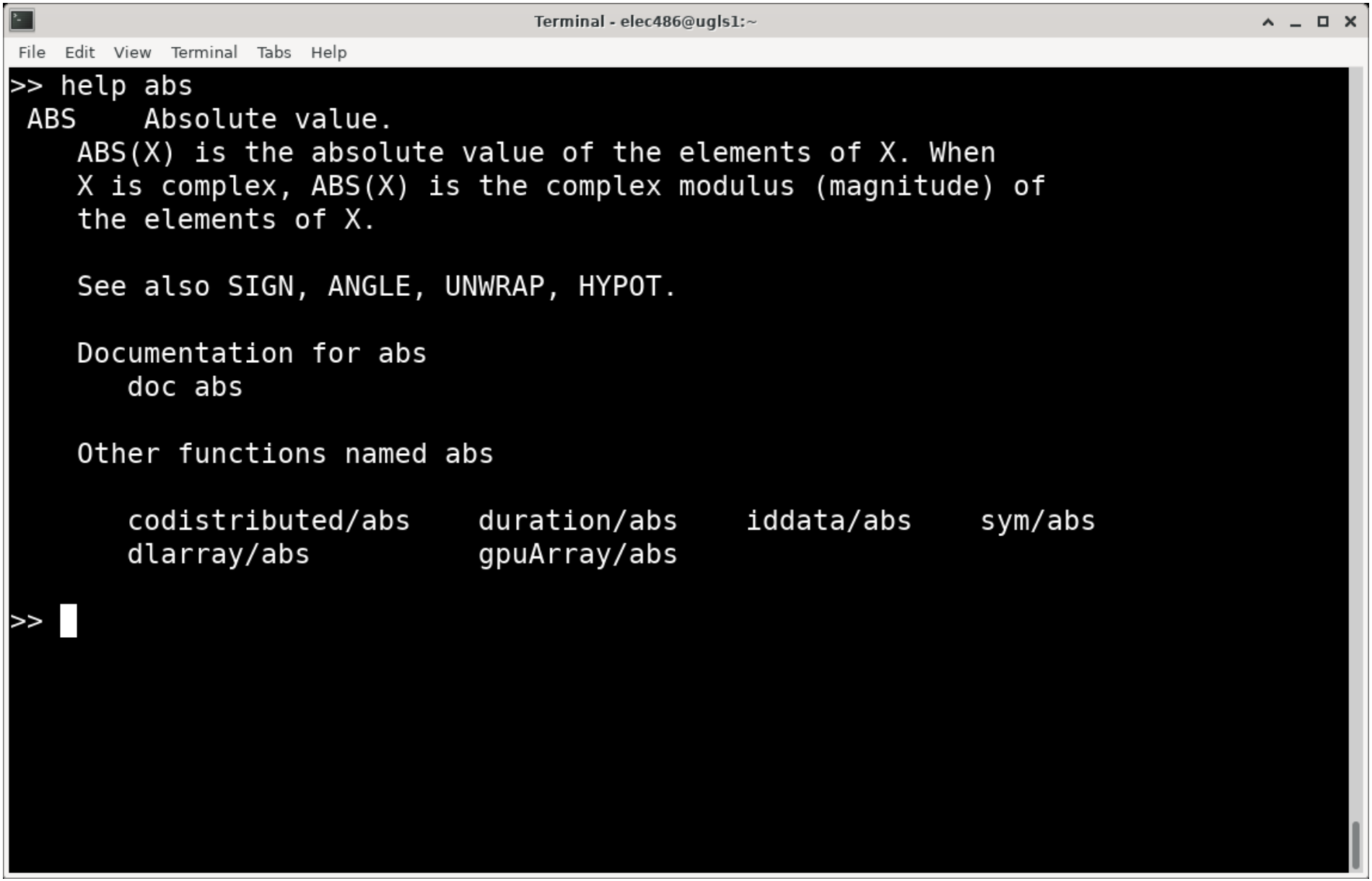
- For example, the following command prints information about the `.*` operator:

```
help .*
```

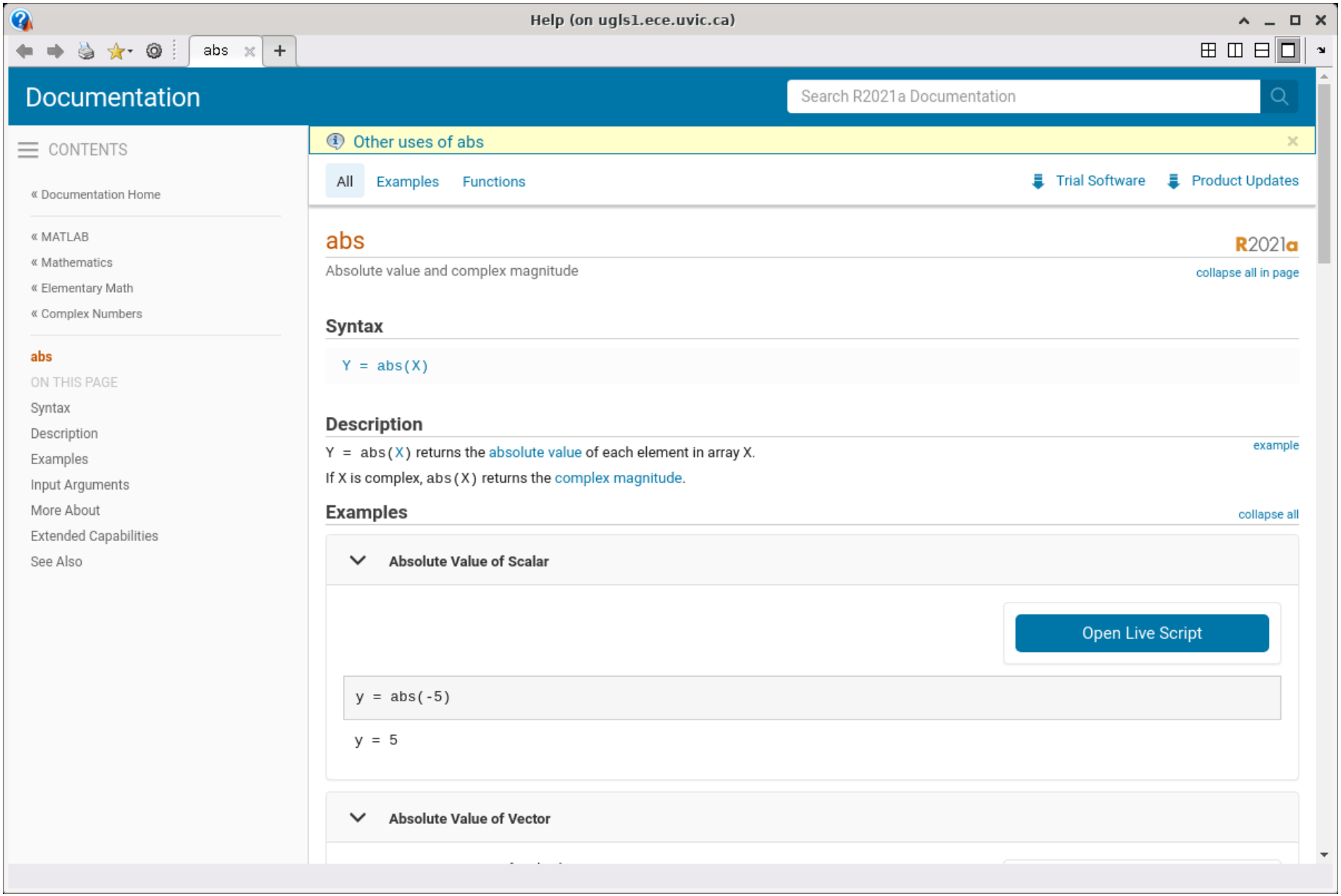
- The `doc` command opens the help browser (if not already open) and uses the browser to display the documentation for the specified item, such as a function, command, operator symbol, method, class, or toolbox.
- For example, the following command displays documentation for the `abs` function in the help browser:

```
doc abs
```

Command-Window Help Information for `abs` Function



Help Browser Information for `abs` Function



Section 15.4

Comments and Variables

Comments

- Two styles of comments are provided:
  1. block comments
  2. short comments
- A block comment:
  - starts with a line that contains a percent character (i.e., "%") immediately followed by a left-brace character (i.e., "{") and has no other characters except spaces and tabs;
  - ends with a line that contains a percent character immediately followed by a right-brace character (i.e., "}") with no other characters except spaces and tabs.
- A short comment starts with a percent character (i.e., "%") that does not start a block comment and continues until the end of line.
- Some examples of comments are given in the following code fragment:

```
% This short comment continues until the end of line.
disp(sin(42)); % Print the sine of 42.
%{
This is a block comment.
This comment spans multiple lines.
%}
```

Identifiers

- An identifier is a name for an entity such as a variable or function.
- Identifiers are case sensitive and may consist of uppercase and lowercase letters, underscores, and digits, but the first character cannot be a digit or an underscore.
- Although an identifier can be arbitrarily long, only the first n characters are significant, where n depends on the particular version of MATLAB being used.
- The `namelengthmax` function can be used to query the precise value of n.
- Some examples of valid and invalid identifiers are given below.

Name	Valid Identifier
`foo`	yes
`foo2`	yes
`f_o_o`	yes
`_foo`	no (due to leading underscore)
`2foo`	no (due to leading digit)
`foo$`	no (due to dollar sign)

Reserved Keywords

- A reserved keyword (which has special meaning to MATLAB) cannot be used as an identifier (e.g., variable or function name).
- The `iskeyword` function can be used to obtain a list of all reserved keywords.
- The reserved keywords in MATLAB include those in the list below.

```
break      end          persistent
case       for          return
catch      function     spmd
classdef   global       switch
continue   if           try
else       otherwise    while
elseif     parfor
```

Semicolons and Commas

- A comma or semicolon separates statements (or parts of statements) within a single line of code.
- For example, an **`if`** statement can be written on a single line using an approach like the following:

```
if x == 0; x = 1; end
```

- In the case of a comma, the result computed from the statement (if any) is printed.
- In the case of a semicolon, the result is not printed.
- In order to avoid overly verbose program output, statements are typically ended by a semicolon instead of a comma.
- For example, the following line of code will not generate any output:

```
x = 1; y = 2 * x;
```

- In contrast, the following line of code will print the values computed for `x` and `y`:

```
x = 1, y = 2 * x,
```

Line Continuation

- Sometimes, it is desirable (for reasons of readability) to split a logical line of the source code across multiple physical lines in a program file.
- Some language constructs, however, must appear on the same logical line of the source code.
- For this reason, the language provides a mechanism for saying that a logical line of source code continues on the next physical line in the program file.
- In particular, an ellipsis consisting of at least three consecutive period (“.”) characters can be used to continue a logical line of the source code on the next physical line in the file.
- Example:

```
x = 1 + 2 + 3 + ...
  4 + 5 + 6;
```

Variables

- A variable does not need to be declared prior to being used.
- A variable simply comes into existence when it is first assigned a value.
- The type of a variable is determined at run time based on the value assigned to the variable.
- Essentially, every variable in MATLAB is deemed to be an array, where a scalar is simply a single-element array.
- Elements of an array are essentially complex numbers, but may be constrained to assume only real or integer values.

Arrays

- Array indices start numbering at 1 (not 0).
- The specification of an array starts with a left (square) bracket and continues to the corresponding right (square) bracket, where these brackets need not be on the same line.
- The elements for each row are specified contiguously.
- Spaces or commas separate elements in the same row of an array.
- Semicolons or newlines separate elements on different rows.
- For example, the following specifies an array with 2 rows and 3 columns, where the first row has the elements 1, 2, and 3, and the second row has the elements 4, 5, and 6:

```
[1 2 3; 4 5 6]
```

- The preceding matrix could also be written as each of the following:

```
[1, 2, 3; 4, 5, 6]
[1 2 3
 4 5 6]
```

Vectors With Equally-Spaced Elements

- MATLAB provides a compact syntax for specifying a vector with equally-spaced elements.
- In particular, an expression of the following form is employed:

  $$start \;:\; step \;:\; end$$

- The above expression is equivalent to a row vector with its first element equal to *start* and each of the subsequent elements increasing in value by *step* (where *step* may be negative) until the value would surpass *end*.
- The notation "*start* `:` `1` `:` *end*" can be further abbreviated to "*start* `:` *end*".
- Some examples of this abbreviated notation are shown below.

Abbreviated Form	Long Form
`1 : 4`	`[1 2 3 4]`
`0 : 0.2 : 1`	`[0 0.2 0.4 0.6 0.8 1]`
`1 : -1 : -2`	`[1 0 -1 -2]`
`0 : 10 : 25`	`[0 10 20]`
`-1.5 : -1 : -4`	`[-1.5 -2.5 -3.5]`

Array Subscripting

- One or more elements of an array x can be accessed by specifying the rows and columns in which the elements are contained using the subscripting operator.
- In particular, $x(rowspec, colspec)$ is the array consisting of the elements of x that are in the rows specified by $rowspec$ and columns specified by $colspec$.
- Here, $rowspec$ is either a vector containing row indices or the special value ":" which means "all rows".
- Similarly, $colspec$ is either a vector containing column indices or the special value ":" which means "all columns".
- Example:
  - Suppose that `x = [1 2 3; 4 5 6; 7 8 9]`.
  - `x(1, 3)` is `3` (i.e., the row 1 column 3 element of `x`).
  - `x(:, 1)` is `[1; 4; 7]` (i.e., the 1st column of `x`).
  - `x(1, :)` is `[1 2 3]` (i.e., the 1st row of `x`).
  - `x(2 : 3, :)` is `[4 5 6; 7 8 9]` (i.e., the 2nd and 3rd rows of `x`).

Array Subscripting By Column-Major Index

- One or more elements of an array x can also be accessed by specifying their index when numbered in column-major order (which orders elements first by column and then by row).
- In particular, $x(indspec)$ is an array that contains the elements of x having the indices specified by $indspec$.
- The dimensions of $x(indspec)$ match the dimensions of $indspec$ if x is a matrix (i.e., not a row or column vector); otherwise, $x(indspec)$ is either a row or column vector to match x.
- Here, $indspec$ is either an array containing element indices or the special value "`:`" which is equivalent to all indices of x specified in a column vector.
- Example:
  - Suppose that `x = [1 4 7; 2 5 8; 3 6 9]` (i.e., $\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$).
  - `x(3)` is `3` (i.e., the 3rd element of `x` in column-major order).
  - `x([1 4 7])` is `[1 4 7]` (i.e., the 1st, 4th, and 7th elements of `x` in column-major order).
  - `x(:)` is `[1; 2; 3; 4; 5; 6; 7; 8; 9]` (i.e., all elements of `x` in column-major order placed in a column vector).

Array-Subscripting Examples

Suppose that `a` is a 10×10 matrix and `x` is 10×1 vector.

Expression	Meaning
`a(1, :)`	first row of `a`
`a(:, 1)`	first column of `a`
`a(1 : 50)`	first 50 elements of `a` arranged in a row vector
`a(1 : 10)`	first 10 elements of `a` arranged in a row vector (i.e., the first column of `a`)
`a(1 : 2 : 10, :)`	odd-indexed rows of `a`
`a(:, 2 : 2 : 10)`	even-indexed columns of `a`
`a(1 : 5, :)`	rows 1 to 5 of `a`
`a(:, 6 : 10)`	columns 6 to 10 of `a`
`a(1 : 2, 9 : 10)`	submatrix consisting of elements that are in rows 1,2 and also in columns 9,10
`x(1 : 3)`	first three elements of `x` (arranged as a row or column vector to match `x`)

Strings

- A string is simply an array whose elements are characters.
- A string literal starts with a single-quote character (“’”) and continues until the next single-quote character that is not part of a pair of consecutive single-quote characters.
- To include a single-quote character in a string, two consecutive single-quotes characters can be used.
- Some examples of string literals are as follows:

```
'Hello, World!'
'This is a test\n'
'Jane''s computer'
```

- The following denotes a string that consists of one single-quote character:

```
''''
```

- For a string `s`, the expression `s(length(s) : -1 : 1)` corresponds to the string reversed (i.e., backwards).

Section 15.5

Operators and Expressions

Operators

- Operators (such as addition and subtraction) are used to perform calculations.
- An operator that takes one operand is said to be **unary**.
- For example, the expression "`-x`" employs the unary minus (i.e., negation) operator, where the minus operator has the single operand `x`.
- An operator that takes two operands is said to be **binary**.
- For example, the expression "`x - y`" employs the binary minus (i.e., subtraction) operator, where the operator has two operands `x` and `y`.

Operators (Continued)

Symbol	Description
`+`	unary plus (i.e., identity) and binary plus (i.e., addition)
`-`	unary minus (i.e., negation) and binary minus (i.e., subtraction)
`*`	multiplication
`/`	right division (`a / b` means `a * b ^ (-1)`)
`\`	left division (`a \ b` means `a ^ (-1) * b`)
`^`	exponentiation
`'`	conjugate transpose
`.*`	elementwise multiplication
`./`	elementwise division
`.^`	elementwise exponentiation
`.'`	transpose

Elementwise Versus Nonelementwise Operators

First Operand	Second Operand	Operator	Result
`[1 -1` `0 1]`	`[1` `3]`	`*`	`[-2` `3]`
`[1 -1` `0 1]`	`[1 2` `3 4]`	`*`	`[-2 -2` `3 4]`
`[1 -1` `0 1]`	`[1 2` `3 4]`	`.*`	`[1 -2` `0 4]`
`[1 -1]`	`[2 2]`	`.*`	`[2 -2]`

Operator Precedence

- When an expression involves multiple operators, the order in which the operators are applied becomes significant.
- Operators are applied in order of decreasing precedence (i.e., priority).
- Operators with the same precedence are evaluated left to right.
- Parentheses can be used to force a particular ordering of operations.
- The precedence of operators in MATLAB is shown in the table below.

Precedence Level	Operators
1 (highest)	`()`
2	`.' ' .^ ^`
3	`unary+ unary- ~`
4	`.* * ./ .\ / \`
5	`binary+ binary-`
6	`:`
7	`< <= > >= == ~=`
8	`&`
9	`|`
10	`&&`
11 (lowest)	`||`

Operator Precedence/Associativity Example

Expression	Fully-Parenthesized Expression
`a + b * x ^ 2`	`a + (b * (x ^ 2))`
`a + b .* x .^ 2`	`a + (b .* (x .^ 2))`
`a + b + c`	`(a + b) + c`
`a > b && c < d || e == f`	`((a > b) && (c < d)) || (e == f)`
`x(m + 1 : 2 * n + 1)`	`x((m + 1) : ((2 * n) + 1))`
`- x * + x`	`(-x) * (+x)`
`0 <= x && x <= 1`	`(0 <= x) && (x <= 1)`
`0 <= x <= 1` (probably bug)	`(0 <= x) <= 1` (which is always `1`)

Relational Operators

- A relational operator is an operator that tests an ordering or equivalence/equality relationship between two values.
- The following relational operators are provided by MATLAB:

Symbol	Description
<	less than
<=	less than or equal
>	greater than
>=	greater than or equal
==	equal
~=	not equal

Logical Operators

- A logical operator is an operator that computes a primitive Boolean function.
- MATLAB considers any nonzero number to be true and zero to be false.
- The following logical operators are provided by MATLAB:

Symbol	Description
&	elementwise AND
|	elementwise OR
~	NOT
&&	short-circuit AND
||	short-circuit OR

- The elementwise AND, elementwise OR, and NOT operators allow nonscalar (i.e., matrix/vector) operands, whereas the short-circuit AND and short-circuit OR operators require scalar operands.

Examples of Expressions Involving Relational/Logical Operators

Variable	Value
`a`	`[0 1 2 3 4]`
`b`	`[4 3 2 1 0]`

Expression	Value
`a == b`	`[0 0 1 0 0]`
`a ~= b`	`[1 1 0 1 1]`
`a > b`	`[0 0 0 1 1]`
`a > 1`	`[0 0 1 1 1]`
`a < b`	`[1 1 0 0 0]`
`a >= 1 & a <= 3`	`[0 1 1 1 0]`
`a < 1 | a > 3`	`[1 0 0 0 1]`
`~a`	`[1 0 0 0 0]`
`0 <= a & a <= 1`	`[1 1 0 0 0]`
`0 <= a <= 1` (probably bug)	`[1 1 1 1 1]` (always all ones)

Short-Circuited Evaluation of Logical Expressions

- The second operand of the short-circuit AND operator (`&&`) is not evaluated if the first operand is false, since the second operand cannot affect the result in this case.
- The second operand of the OR operator (`||`) is not evaluated if the first operand is true, since the second cannot affect the result in this case.
- The above behavior is known as short-circuit evaluation.
- Example:
  - Suppose that `x` and `y` are initialized as `x = 0` and `y = 1`.
  - If MATLAB is asked to evaluate the logical expression `x ~= 0 && y < 1 / x`, only `x ~= 0` will be evaluated, since the result must be false regardless of the value of `y < 1 / x`.
  - Note that short circuit evaluation guarantees division by zero cannot occur during the evaluation of `x ~= 0 && y < 1 / x`.
  - If MATLAB is asked to evaluate the logical expression `x ~= 42 || y > 0`, only `x ~= 42` will be evaluated, since the result must be true regardless of the value of `y > 0`.

Section 15.6

Selection Constructs

`if` Statements

- An **`if`** statement allows groups of statements to be conditionally executed, and has a number of variants.
- The simplest variant (i.e., the **`if`** variant) has the form:

  `if` *expression*
  statements
  `end`

- If the expression *expression* is true, then the statements *statements* are executed.
- Example:

```
if x < 0
   disp('x is negative');
end
```

`if` Statements (Continued 1)

- The next simplest variant of the **`if`** statement (i.e., the **`if`**–**`else`** variant) has the form:

  `if` $expression_1$
  $statements_1$
  `else`
  $statements_2$
  `end`

- If the expression $expression_1$ is true, then the statements $statements_1$ are executed; otherwise, the statements $statements_2$ are executed.
- Example:

```
if x < 0
    disp('x is negative');
else
    disp('x is not negative');
end
```

`if` Statements (Continued 2)

- Finally, the most general variant of the **`if`** statement has the following form (where the **`elseif`** and **`else`** clauses are optional):

`if` $expression_1$
  $statements_1$
`elseif` $expression_2$
  $statements_2$
  $\vdots$
`elseif` $expression_{n-1}$
  $statements_{n-1}$
`else`
  $statements_n$
`end`

- For each **`if`**/**`elseif`** clause in order of appearance, the expression $expression_i$ is evaluated, and if this expression is true, the statements $statements_i$ are executed and the remainder of the **`if`** statement is skipped.
- If none of the expressions in the **`if`**/**`elseif`** clauses are true and an **`else`** clause is present, the statements $statements_n$ in this clause are executed.

`if` Statement Example

```
1  % Read a real number and then print information
2  % about its sign.
3  x = input('Enter a real number: ');
4  if x > 0
5     disp('x is strictly positive');
6  elseif x < 0
7     disp('x is strictly negative');
8  else
9     disp('x is zero');
10 end
```

`switch` Statements

- The **`switch`** statement provides another means to conditionally execute groups of statements.
- The general form of a **`switch`** statement is as follows (where the **`otherwise`** clause is optional):

`switch` *expression*
`case` $test\_expression_1$
$statements_1$
`case` $test\_expression_2$
$statements_2$
$\vdots$
`case` $test\_expression_{n-1}$
$statements_{n-1}$
`otherwise`
$statements_n$
`end`

- The first test expression, say $test\_expression_k$, matching the expression *expression* has its corresponding statements $statements_k$ executed.
- If none of the test expressions match the expression *expression* and an **`otherwise`** clause is present, the statements $statements_n$ in this clause are executed.
- The switch expression *expression* must be either a scalar or string.

`switch` Statement Example [Scalar]

```
1  % Get a real value and print some information about it.
2  n = input('Enter real number: ');
3  switch mod(n, 2)
4  case 0
5     disp('number is even integer');
6  case 1
7     disp('number is odd integer');
8  case {0.5, 1.5}
9     disp('number is odd multiple of one half');
10 otherwise
11    disp('number is not multiple of one half');
12 end
```

`switch` Statement Example [String]

```
% Get the type of a polygon and then print the number of
% sides that the polygon has.
type = input('Enter polygon type: ', 's');
switch lower(type)
case 'triangle'
    num_sides = 3;
case 'quadrilateral'
    num_sides = 4;
case 'pentagon'
    num_sides = 5;
case 'hexagon'
    num_sides = 6;
case 'heptagon'
    num_sides = 7;
case 'octagon'
    num_sides = 8;
case 'nonagon'
    num_sides = 9;
case 'decagon'
    num_sides = 10;
otherwise
    error('unknown polygon type');
end
fprintf('The polygon has %d sides.\n', num_sides);
```

Section 15.7

Looping Constructs

`for` Statements

- The **`for`** statement allows a group of statements to be repeated a fixed number of times.
- A **`for`** statement has the general form:

  `for` *variable* = *array*
  statements
  `end`

- The statements *statements* are executed once for each column in the array *array*, where the variable *variable* is set to the corresponding array column each time.
- The following code fragment prints the integers 1, 3, 7, and 9:

```
for k = [1 3 7 9]
    disp(k);
end
```

`for` Statement Example

Code

```
1  fprintf('Degrees  Radians\n');
2  for theta_deg = -180 : 30 : 180
3     theta_rad = theta_deg * pi / 180;
4     fprintf('%7.1f  %7.4f\n', theta_deg, theta_rad);
5  end
```

Output

```
   Degrees     Radians
    -180.0     -3.1416
    -150.0     -2.6180
    -120.0     -2.0944
     -90.0     -1.5708
     -60.0     -1.0472
     -30.0     -0.5236
       0.0      0.0000
      30.0      0.5236
      60.0      1.0472
      90.0      1.5708
     120.0      2.0944
     150.0      2.6180
     180.0      3.1416
```

`while` Statements

- The **`while`** statement allows a group of statements to be executed an indefinite number of times.
- A **`while`** statement has the general form:

  `while` *expression*
  statements
  `end`

- The statements *statements* are executed repeatedly as long as the condition *expression* is true.
- The following code fragments prints random numbers obtained via the `rand` function until a number not exceeding 0.1 is obtained:

```
x = rand;
while x > 0.1
    disp(x);
    x = rand;
end
```

`break` Statements

- Sometimes, it may be necessary to terminate a loop by code executing in the body of the loop.
- This can be accomplished using a **`break`** statement.
- Typically, a **`break`** statement is used to terminate a loop from code that is somewhere in the middle of the loop body.
- The following code uses a **`break`** statement in order to terminate a **`while`** loop when a particular condition is satisfied:

```
values = [];
while 1
    x = rand;
    if x > 0.9
        break
    end
    values = [values x];
end
disp(values);
```

`continue` Statements

- Sometimes, it may necessary to start the next iteration of a loop without completing the remainder of the code in the loop body.
- This is accomplished using a **`continue`** statement.
- The following code fragment uses a **`continue`** statement to skip over the processing of any zero elements in the array `a`:

```
a = [1 0 3 2 0];
for i = a
    if i == 0
        % Skip over the processing of a
        % zero element in the array.
        continue
    end
    % Process the nonzero array element.
    disp(i);
end
```

Section 15.8

Using Functions

Calling Functions

- A function that can be called with no arguments can be invoked by specifying either its name or its name followed by a pair of empty parentheses (with the former usually being preferred since it is less verbose).
- For example, to invoke the built-in `j` function (which takes no arguments and returns $\sqrt{-1}$), we would write either "`j`" or "`j()`".
- A function that requires one or more arguments is called by placing the list of arguments to the function (separated by commas) in parentheses (i.e., round brackets) after the name of the function.
- Examples:
  - To invoke the `sin` function with the single argument `42`, we would write "`sin(42)`".
  - To invoke the `atan2` function with the arguments `y` and `x`, we would write "`atan2(y, x)`".

Built-In Functions

- MATLAB has many built-in (i.e., automatically predefined) functions.
- Although a built-in function can be overridden by a user-defined function or variable, doing this can often lead to confusing code.
- Consequently, it is recommended that the overriding of built-in functions be avoided in most cases.
- One exception to this rule might be overriding the built-in `i` and `j` functions in code that does not use complex arithmetic, since this is unlikely to lead to confusion and `i` and `j` are often desirable names for looping variables.

Some Built-In Functions

Name	Description
`pi`	π
`i`	$\sqrt{-1}$
`j`	$\sqrt{-1}$
`nan`	not-a-number (NaN)
`inf`	infinity
`ans`	last expression evaluated that was not assigned to variable
`date`	date
`clock`	wall clock
`realmin`	smallest usable positive real number
`realmax`	largest usable positive real number

Built-In Math Functions

Elementary Math Functions

Name	Description
`abs`	magnitude of complex number
`angle`	principal argument of complex number
`imag`	imaginary part of complex number
`real`	real part of complex number
`conj`	conjugate of complex number
`sign`	signum function
`rem`	remainder (with same sign as dividend)
`mod`	remainder (with same sign as divisor)

Rounding Functions

Name	Description
`round`	round to nearest integer
`fix`	round towards zero
`floor`	round towards $-\infty$
`ceil`	round towards ∞

Built-In Math Functions (Continued 1)

Basic Statistical Functions

Name	Description
`min`	minimum value
`max`	maximum value
`mean`	mean value
`std`	standard deviation
`median`	median value

Sum and Product Functions

Name	Description
`sum`	sum of elements
`prod`	product of elements
`cumsum`	cumulative sum of elements
`cumprod`	cumulative product of elements

Built-In Math Functions (Continued 2)

Exponential and Logarithmic Functions

Name	Description
`exp`	exponential function
`log`	natural logarithmic function
`log10`	base-10 logarithmic function
`sqrt`	square root function

Trigonometric Functions

Name	Description
`sin`	sine function
`cos`	cosine function
`tan`	tangent function
`asin`	arcsine function
`acos`	arccosine function
`atan`	arctangent function
`atan2`	two-argument form of arctangent function

Radix Conversion Functions

Name	Description
`dec2bin`	convert decimal to binary
`bin2dec`	convert binary to decimal
`dec2hex`	convert decimal to hexadecimal
`hex2dec`	convert hexadecimal to decimal
`dec2base`	convert decimal to arbitrary radix
`base2dec`	convert arbitrary radix to decimal

Other Math Functions

Name	Description
`sinc`	***normalized*** sinc function $\left(\text{i.e., } f(t) = \frac{\sin(\pi t)}{\pi t}\right)$
`polyval`	evaluate polynomial
`cart2pol`	Cartesian-to-polar coordinate conversion
`pol2cart`	polar-to-Cartesian coordinate conversion

Matrix Computation Functions

Name	Description
`expm`	compute matrix exponential
`det`	compute determinant of matrix
`inv`	compute inverse of matrix
`eig`	compute eigenvalues and eigenvectors of matrix
`svd`	compute singular-value decomposition (SVD) of matrix
`lu`	compute lower-upper (LU) decomposition of matrix
`qr`	compute QR decomposition of matrix

Functions Related to Array Size

Name	Description
`size`	get row vector containing size of array in each dimension
`width`	get number of columns in array
`height`	get number of rows in array
`length`	get size of largest array dimension (e.g., for vector, yields vector's length)
`ndims`	get number of array dimensions (always at least two)
`numel`	get number of elements in array
`isrow`	tests if array is row vector
`iscolumn`	tests if array is column vector
`isvector`	tests if array is row or column vector
`ismatrix`	tests if array is matrix
`isempty`	tests if array is empty (i.e., contains no elements)

Array-Size Functions Example

```
1  x = input('Enter a matrix/vector: ');
2  if numel(x) < 2
3     error('invalid matrix/vector (i.e., scalar or empty)');
4  end
5  if isrow(x)
6     disp('array is row vector');
7  elseif iscolumn(x)
8     disp('array is column vector');
9  elseif ismatrix(x)
10    disp('array is matrix');
11 end
12 fprintf('array contains %d elements\n', numel(x));
13 fprintf('array elements in row-major order:\n');
14 for i = 1 : height(x)
15    for j = 1 : width(x)
16       fprintf("%.15f\n", x(i, j));
17    end
18 end
```

Built-In Array Functions

Special Matrix/Vector Functions

Name	Description
`eye`	identity matrix
`ones`	matrix of ones
`zeros`	matrix of zeros
`diag`	diagonal matrix
`rand`	random matrix
`linspace`	vector with linearly spaced elements
`logspace`	vector with logarithmically spaced elements

Basic Array Manipulation Functions

Name	Description
`rot90`	rotate array by 90 degrees
`fliplr`	flip array horizontally
`flipud`	flip array vertically
`reshape`	change array dimensions

Array-Functions Example

Expression	Result
`zeros(2, 3)`	`[0 0 0` `0 0 0]`
`ones(2, 3)`	`[1 1 1` `1 1 1]`

`eye(3)`	`[1 0 0`
	`0 1 0`
	`0 0 1]`

`linspace(0, 4, 5)`	`[0 1 2 3 4]`

Relational and Logical Functions

Name	Description
`any`	any element nonzero
`all`	all elements nonzero
`find`	find nonzero elements
`exist`	check if variable/function exists
`isfinite`	detect finite values
`isinf`	detect infinities
`isnan`	detect NaNs
`isempty`	detect empty matrices
`isstr`	detect strings
`strcmp`	compare strings

Logical Functions Example

```
a = [1 2 3; 4 5 6; 7 8 9];
if all(a > 0)
   disp('All matrix elements are positive.');
end
if ~any(a == 0)
   disp('All matrix elements are nonzero.');
end
if all(real(a) == a)
   disp('All matrix elements are real.');
end
i = find(a >= 8);
disp(['The following matrix elements are ', ...
  'greater than or equal to 8:']);
disp(a(i));
```

Section 15.9

Scripts and Functions

M Files

- MATLAB code can be placed in a file, called an M file, for later execution.
- An M file must have file name extension "`.m`" (which is where term "M file" originates).
- M files are used for scripts and specifying user-defined functions.
- Depending on how the code in an M file is structured it can correspond to either a script or user-defined function.
- By using the MATLAB identifier *name*, one can execute the code stored in an M file named "*name*`.m`".
- For example, the MATLAB identifier `my_demo` would refer to the MATLAB code in the M file named `my_demo.m` (assuming that the identifier `my_demo` is not used as a variable name).

Scripts

- An M file that does not define a function is referred to as a script.
- Executing code in a script behaves as if the contents of the script were pasted into the place where script is being referenced.
- Since a script is executed in the context of its invoker, any changes made to variables are visible to the invoker.

Script Example: `greet`

Contents of file `greet.m`

```
1  % Print a greeting.
2  disp('Hello, World.');
```

- The code in the file `greet.m` shown above can be executed by simply typing the following in the MATLAB command window:

  ```
  greet
  ```

- Executing the code will result in the `disp` function being invoked to print the message "`Hello, World!`".

Script Example: demo

Contents of file demo.m

```
1  t = [0 : .1 : 8];
2  f = sin(t);
3  plot(t, f);
```

- The code in the file demo.m shown above can be executed by simply typing the following in the MATLAB command window:

  ```
  demo
  ```

- Recall that a script is executed in the context of the invoker so any changes made to variables by the script are visible to the invoker.
- For example, if the code in the above script is invoked as follows, the changes to the variables t and f are visible to the invoker:

  ```
  clear t f % Make t and f undefined.
  demo % execute code in demo.m script
  disp(t); % OK: t was defined by script
  disp(f); % OK: f was defined by script
  ```

MATLAB Search Path

- The MATLAB search path is a list of directories that MATLAB examines in order to locate various files, such as MATLAB program files (i.e., M files) and files associated with toolboxes.
- When trying to locate a file, MATLAB will look for the file in each directory in the search path in order until the file is found.
- The order in which directories appear in the search path is important, since the directories are searched in that order.
- Normally, the search path contains the current working directory (which appears first) as well as any directories associated with files that are part of the MATLAB software installation.

How MATLAB Resolves Identifiers

- When MATLAB encounters an identifier, it must decide the specific entity to which the identifier refers (such as a variable, built-in function, or script or function specified in an M file).
- To do this for the identifier *id*, MATLAB employs an algorithm consisting of the following steps (in order):
  1. If a ***user-defined (i.e., not built-in) variable*** exists with *id* as its name, *id* is deemed to refer to this variable.
  2. Otherwise, if *id* has a corresponding M file (i.e., a file named *id*`.m`) ***in current directory***, *id* is deemed to refer to the function or script associated with that M file.
  3. Otherwise, if *id* has a corresponding M file (i.e., a file named *id*`.m`) in any directory ***on the MATLAB search path***, *id* is deemed to refer to the function or script associated with the first such M file found on this path.
  4. Otherwise, if a ***built-in*** variable/function exists with *id* as its name, *id* is deemed to refer to this variable/function.
- If multiple entities are associated with the same identifier *id*, the order of the above steps becomes important in deciding which entity will be referenced by MATLAB when *id* is used.

Naming Conflicts

- Generally, it is advisable to avoid naming variables and M files in ways that could result in conflicts.
- When multiple entities are associated with the same identifier, one entity can hide (i.e., override) another.
- The use of an identifier that would conflict with any built-in function should be avoided in most cases.
- A user-defined variable will hide a function of the same name. For example:

```
sin = 42
% can no longer call sin function
sin(pi) % ERROR: trying to subscript with index pi
```

- An M file in the current directory will hide all other M files of the same name in the MATLAB search path.

MATLAB Search Path

- All of the directories in the MATLAB search path (i.e., all of the directories in which MATLAB will look for M-file scripts) can be printed with the `path` command:

  ```
  path
  ```

- The working directory for MATLAB can be changed to *dirname* using the `cd` command:

  ```
  cd dirname
  ```

- The current working directory can be printed with the `pwd` command:

  ```
  pwd
  ```

Functions

- A function definition is introduced using the following syntax:

$$\texttt{function}\ [out_1, out_2, \ldots, out_n] = func(in_1, in_2, \ldots, in_m)$$

- This indicates that the function has:
  - the name *func*,
  - the m input arguments $in_1, in_2, \ldots, in_m$, and
  - the n output arguments $out_1, out_2, \ldots, out_n$.
- The body of a function extends from the **`function`** directive to a corresponding **`end`** directive or, if no such **`end`** directive is present, the end of the file.
- The code in a function executes until either the end of the function is reached or a **`return`** statement is encountered.
- In order for the function *func* to be callable from code outside the source file containing the function, the function must be placed in a file named "*func*`.m`" and be the first function defined in that file.

Functions (Continued)

- Immediately following the line containing the **`function`** directive, one should provide comments to be printed in response to a `help` inquiry for the function.
- In MATLAB all input arguments to a function are passed by value.
- For this reason, changes to the input arguments made inside of a function will not propagate to the caller.
- Also, any variables accessed/manipulated inside of a function are local in scope to that function.

Function Scoping Example

demo.m

```
1  function demo
2      t = [0 : .1 : 8];
3      f = sin(t);
4      plot(t, f);
5  end
```

Code to Invoke `demo` Function (e.g., in Command Window)

```
1  clear t f % Make t and f undefined.
2  demo
3  disp(t); % ERROR: t undefined
4  disp(f); % ERROR: f undefined
```

- The above is identical to an earlier code example for scripts, except that the code in `demo.m` has been moved into the body of the function `demo`.
- The above example illustrates a key difference between invoking scripts versus invoking functions, namely: a function introduces a new scope (i.e., variables defined/accessed in a function are local to that function).

Function Example: my_sqr

my_sqr.m

```
1  function y = my_sqr(x)
2      % Compute the square of each element in an array.
3
4      % Use the elementwise product so that the code
5      % works in both the scalar and non-scalar cases.
6      y = x .^ 2;
7  end
```

my_sqr_main.m

```
1  a = my_sqr([1 2 3]);
2  disp(a);
```

Command Window Output

```
>> my_sqr_main
     1     4     9

>>
```

Function Example: fahrenheit_to_celsius

fahrenheit_to_celsius.m

```
1  function c = fahrenheit_to_celsius(f)
2      % Convert Fahrenheit to Celsius.
3      c = (5 / 9) * (f - 32);
4  end
```

fahrenheit_to_celsius_main.m

```
1  fprintf('%10s %10s\n', 'Fahrenheit', 'Celsius');
2  for f = 30 : 10 : 100
3      c = fahrenheit_to_celsius(f);
4      fprintf('%10.2f %10.2f\n', f, c);
5  end
```

Command Window Output

```
>> fahrenheit_to_celsius_main
Fahrenheit   Celsius
     30.00     -1.11
     40.00      4.44
     50.00     10.00
     60.00     15.56
     70.00     21.11
     80.00     26.67
     90.00     32.22
    100.00     37.78
>>
```

Function Example: roots_of_quadratic

roots_of_quadratic.m

```
1  function [r1 r2] = roots_of_quadratic(a, b, c)
2      % Find the two roots of the quadratic equation
3      % a * x ^ 2 + b * x + c = 0.
4      if a == 0
5          error('leading coefficient is zero');
6      end
7      d = sqrt(b ^ 2 - 4 * a * c);
8      r1 = (-b - d) / (2 * a);
9      r2 = (-b + d) / (2 * a);
10 end
```

roots_of_quadratic_main.m

```
1  [a b] = roots_of_quadratic(1, 0, -16);
2  fprintf('The roots are %f and %f.\n', a, b);
```

Command Window Output

```
>> roots_of_quadratic_main
The roots are -4.000000 and 4.000000.
>>
```

Function Example: my_mean

my_mean.m

```
1  function y = my_mean(x)
2      % Get the average of the elements in an array.
3      y = sum(x(:)) / numel(x);
4  end
```

my_mean_main.m

```
1  x = [1 5 7 3 8 4];
2  y = my_mean(x);
3  fprintf('The mean is %f.\n', y);
```

Command Window Output

```
>> my_mean_main
The mean is 4.666667.
>>
```

Function Example: my_factorial [Multiple Functions Per File]

my_factorial.m

```
1   function f = my_factorial(n)
2       % Compute the factorial of each element of an array.
3       f = arrayfun(@my_factorial_scalar, n);
4   end
5
6   function f = my_factorial_scalar(n)
7       % Compute the factorial of a single number.
8       if floor(n) ~= n || n < 0
9           error('nonnegative integer required');
10      end
11      f = 1;
12      while n >= 2
13          f = f * n;
14          n = n - 1;
15      end
16  end
```

my_factorial_main.m

```
1   f = my_factorial(1 : 5);
2   fprintf('%d ', f); fprintf('\n');
```

Command Window Output

```
>> my_factorial_main
1 2 6 24 120
>>
```

Variadic Functions

- In MATLAB, functions may take a variable number of input arguments and may return a variable number of output arguments.
- In order for a function to determine the number of input and output arguments and access these arguments, several variables are automatically defined upon entry to a function, including those listed below.

Name	Description
`nargin`	number of input arguments
`nargout`	number of output arguments
`varargin`	variable-length input argument
`varargout`	variable-length output argument

Variadic Function Example: mysum

mysum.m

```
1   function y = mysum(a, b, c)
2       % mysum - calculate the sum (of one to three quantities)
3       if nargin == 1
4           % function called with one argument
5           y = a;
6       elseif nargin == 2
7           % function called with two arguments
8           y = a + b;
9       elseif nargin == 3
10          % function called with three arguments
11          y = a + b + c;
12      else
13          error('invalid number of arguments');
14      end
15  end
```

mysum_main.m

```
1   fprintf('%f %f %f\n', mysum(1), mysum(1, 2), mysum(1, 2, 3));
```

Command Window Output

```
>> mysum_main
1.000000 3.000000 6.000000
>>
```

Variadic Function Example: mysum2

mysum2.m

```
function y = mysum2(varargin)
    % mysum2 - Compute the sum of the input arguments.
    if nargin == 0
        y = [];
        return
    end
    y = varargin{1};
    for i = 2 : nargin
        y = y + varargin{i};
    end
end
```

mysum2_main.m

```
fprintf('%f %f\n', mysum2(1, 1, 1, 1, 1, 1, 1, 1), ...
  mysum(1, 2));
```

Command Window Output

```
>> mysum2_main
8.000000 3.000000
>>
```

Function Handles and Anonymous Functions

- A function handle provides a means by which to refer to a function.
- A handle to the function f is specified by the syntax `@`f.
- For example:

```
func = @sin;
x = func(42); % set x to sin(42)
```

- An anonymous function is a function with no name that is identified only by a function handle and is specified using a special syntax.
- This syntax specifies the function parameters and the function body all in a single statement.
- An anonymous function is introduced by an at-character (i.e., "`@`") followed by the parameter list for the function enclosed in parentheses followed by the single-statement function body.
- If an anonymous function is assigned to a variable, that variable can then be used as a way invoke the anonymous function.
- For example:

```
square = @(x) x .^ 2;
x = square(2); % set x to 2 .^ 2
```

Sinc Function

- The built-in MATLAB function `sinc` computes the ***normalized sinc function*** (i.e., $f(t) = \frac{\sin(\pi t)}{\pi t}$), not the sinc function (i.e., $\text{sinc}(t) = \frac{\sin(t)}{t}$).
- With some care to avoid division by zero, the (unnormalized) sinc function can be implemented as the MATLAB function `usinc` as follows:

```
function f = usinc(t)
    f = ([t == 0] + sin(t)) ./ ([t == 0] + t);
end
```

- The above `usinc` function could be implemented more compactly using an anonymous function as follows:

```
usinc = @(t) ([t == 0] + sin(t)) ./ ([t == 0] + t);
```

- Of course, the above `usinc` function could also be implemented in terms of MATLAB's built-in `sinc` function as follows:

```
usinc = @(t) sinc(t / pi)
```

Section 15.10

Data Visualization

Data Visualization and Plotting

- MATLAB has extensive data-visualization capabilities with support for many types of plots, including:
  - line plots in 2-D and 3-D (with support for such things as error bars, stacked plots, and staircase plots)
  - scatter plots and bubble charts in 2-D and 3-D
  - histograms and pie charts in 2-D and 3-D and heat maps
  - discrete plots (such as stem, bar, and staircase plots) in 2-D and 3-D
  - geographic plots
  - polar plots
  - contour plots
  - quiver plots in 2-D and 3-D
  - surface and mesh plots
  - volume visualization (e.g., streams)
  - animated plots
  - images

Figures and Plotting

- Data visualization output such as plots are generated in figure windows.
- A new figure window can be created with the `figure` function.
- A figure window is partitioned into an $m \times n$ rectangular grid of subfigures, where a subfigure is simply an individual data visualization, such as a plot.
- When a figure is first created, m and n are effectively defaulted to 1.
- The values of m and n can be changed with the `subplot` function.
- The `subplot` function is also used to control which subfigure should be used when drawing new data visualizations (e.g., plots).
- Normally, plotting data in a subfigure first erases any data already being shown in the subfigure.
- The `hold` function can be used to override this behavior so that new data can be added to an already existing plot (e.g., to allow plotting multiple lines on a single plot).

Plotting-Related Functions

Basic 2-D Plotting Functions

Name	Description
`plot`	linear x-y plot
`loglog`	log log x-y plot
`semilogx`	semi-log x-y plot (x-axis logarithmic)
`semilogy`	semi-log x-y plot (y-axis logarithmic)
`polar`	polar plot
`bar`	bar chart
`stem`	stem plot
`pcolor`	pseudocolor (checkerboard) plot
`fplot`	plot function associated with function handle

Other Plotting-Related Functions

Name	Description
`axis`	control axis scaling and appearance
`hold`	hold current plot
`subplot`	multiple axes in single figure
`figure`	create figure

Plotting-Related Functions

Plot Annotation Functions

Name	Description
`title`	graph title
`xlabel`	x-axis label
`ylabel`	y-axis label
`grid`	grid lines
`text`	arbitrarily-positioned text
`gtext`	mouse-positioned text

Figure Saving/Printing Functions

Name	Description
`print`	print figure to file or device
`saveas`	save figure to file

3-D Plotting Functions

Name	Description
`plot3`	plot lines/points in 3-D space
`surf`	plot surface in color

Symbols for Line Styles, Line Colors, and Marker Styles

Line Styles

Symbol	Line Style
`-`	solid
`:`	dotted
`-.`	dash dot
`--`	dashed

Line Colors

Symbol	Line Color
`b`	blue
`g`	green
`r`	red
`c`	cyan
`m`	magenta
`y`	yellow
`k`	black
`w`	white

Marker Styles

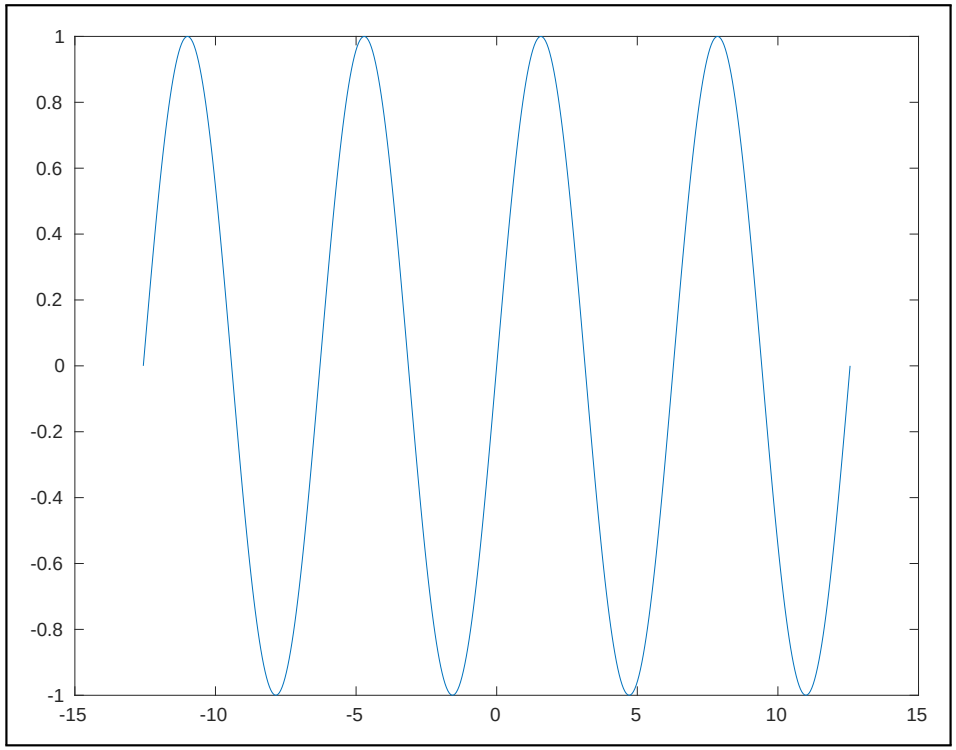
Symbol	Marker Style
`.`	point
`o`	circle
`x`	cross
`+`	plus sign
`*`	asterisk
`s`	square
`d`	diamond
`v`	triangle (down)
`^`	triangle (up)
`<`	triangle (left)
`>`	triangle (right)
`p`	pentagram
`h`	hexagram

Special Symbols for Plot Annotations

String	Symbol
\alpha	α
\beta	β
\delta	δ
\gamma	γ
\omega	ω
\theta	θ
\Delta	Δ
\Omega	Ω

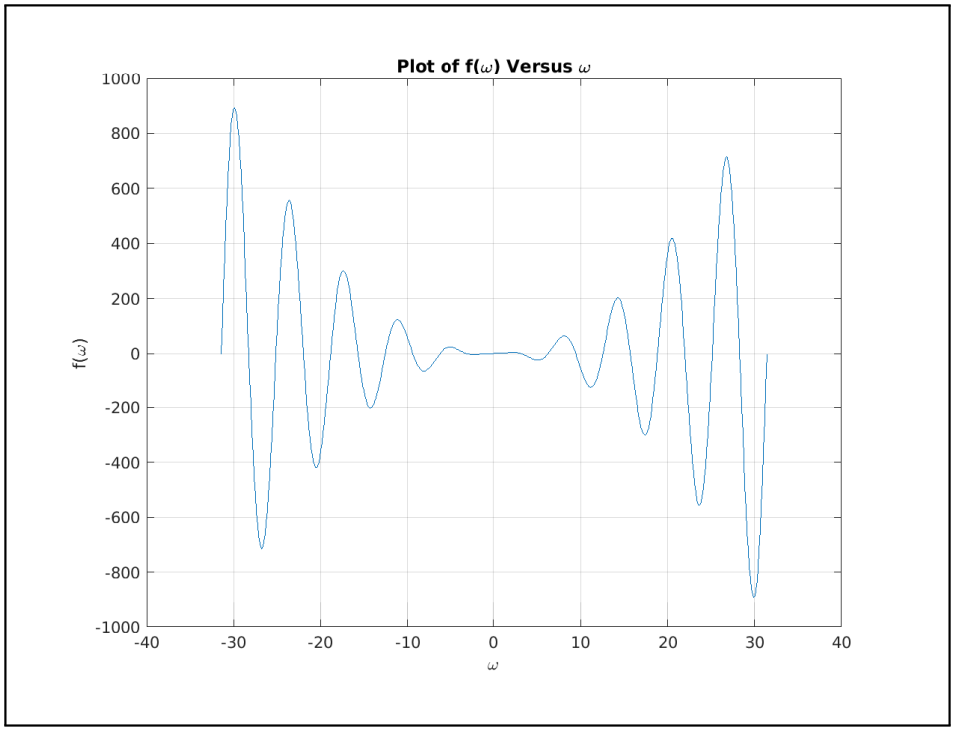
Example: Simple Line Plot [figure, plot]

```
1  figure
2  t = linspace(-4 * pi, 4 * pi, 500);
3  y = sin(t);
4  plot(t, y);
5  print('output.pdf', '-dpdf', '-bestfit');
```



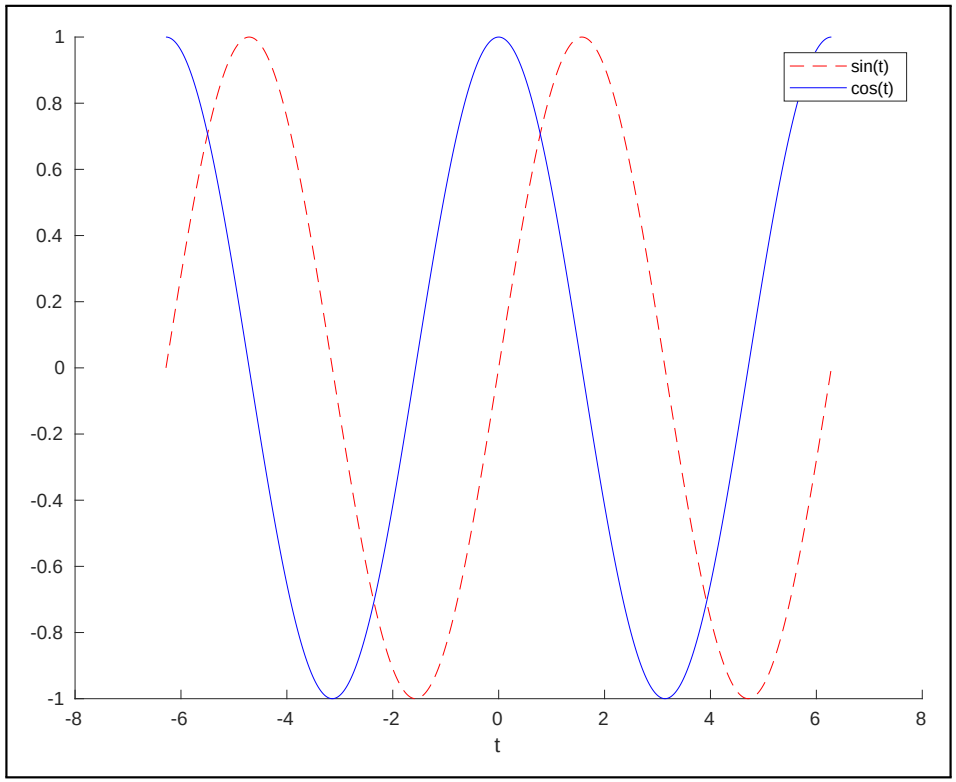
Example: Annotated Line Plot [title, xlabel, ylabel, grid]

```
figure
w = linspace(-10 * pi, 10 * pi, 500);
f = abs(w) .^ 2 .* sin(w);
plot(w, f);
title('Plot of f(\omega) Versus \omega');
xlabel('\omega');
ylabel('f(\omega)');
grid on
print('output.pdf', '-dpdf', '-bestfit', '-opengl');
```



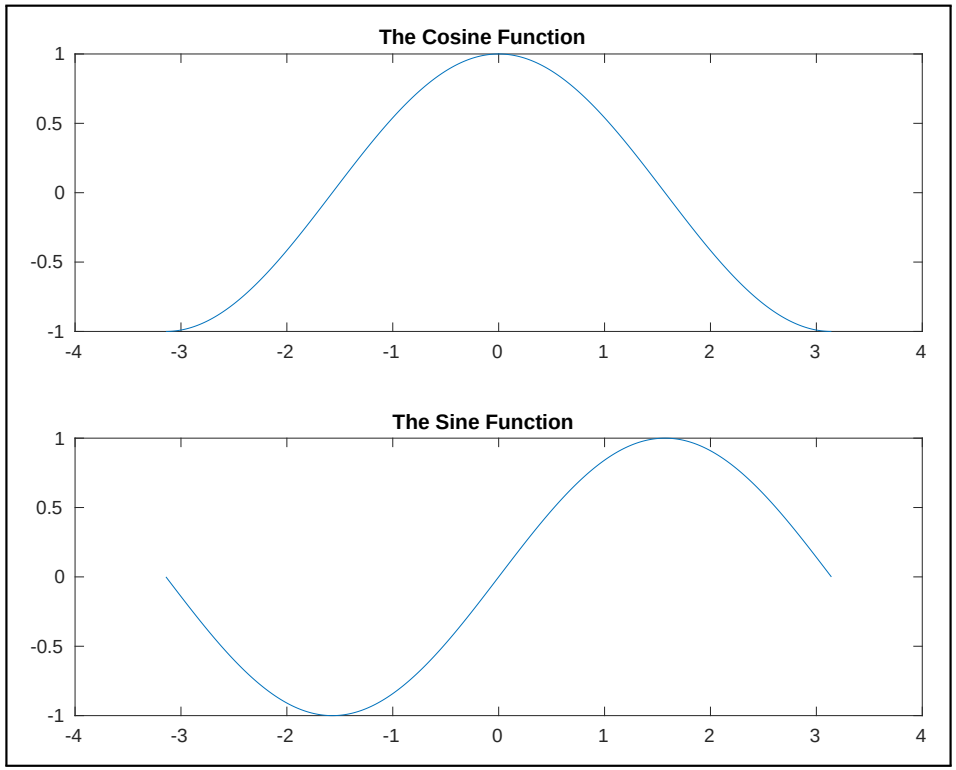
Example: Multiple Lines Per Plot [hold, legend]

```
1  figure
2  t = linspace(-2 * pi, 2 * pi, 500);
3  hold on
4  plot(t, sin(t), 'r--');
5  plot(t, cos(t), 'b-');
6  hold off
7  xlabel('t');
8  legend('sin(t)', 'cos(t)');
9  print('output.pdf', '-dpdf', '-bestfit');
```



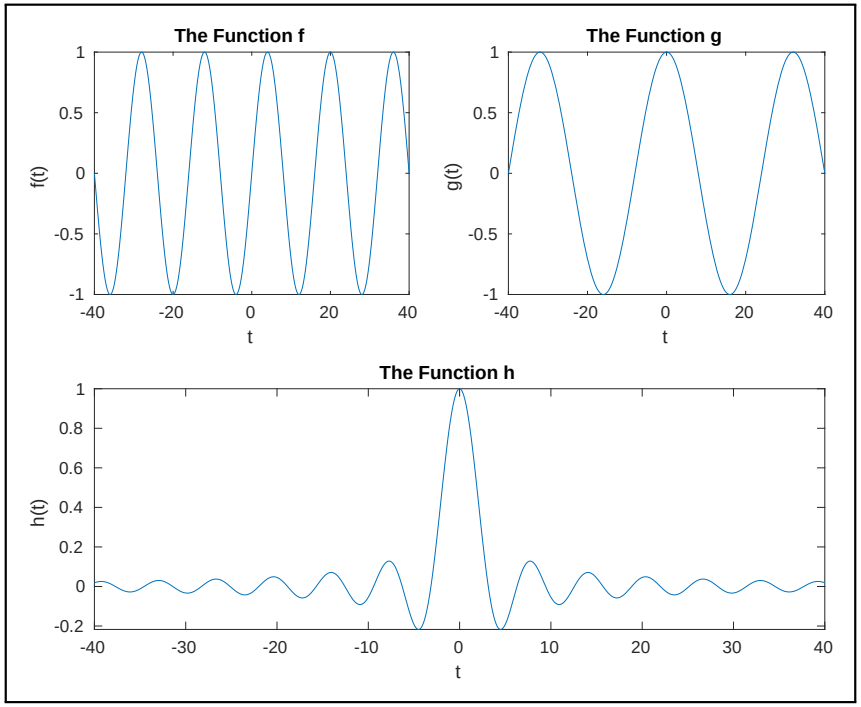
Example: Multiple Plots Per Figure [subplot]

```
1   figure
2   t = linspace(-pi, pi, 500);
3   subplot(2, 1, 1);
4   plot(t, cos(t));
5   title('The Cosine Function')
6   subplot(2, 1, 2);
7   plot(t, sin(t));
8   title('The Sine Function');
9   print('output.pdf', '-dpdf', '-bestfit');
```



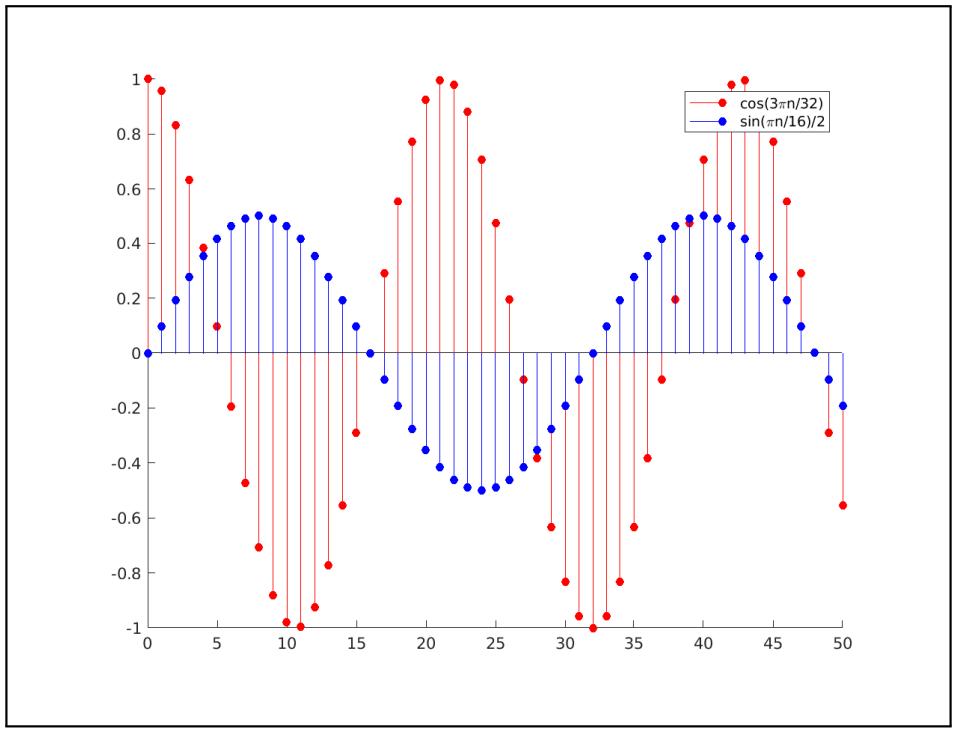
Example: Multiple Plots Per Figure [Different `subplot` Grids]

```
 1   figure
 2   t = [-40 40];
 3   subplot(2, 2, 1);
 4   fplot(@(t) sin(1 / 8 * pi * t), t);
 5   title('The Function f'); xlabel('t'); ylabel('f(t)');
 6   subplot(2, 2, 2);
 7   fplot(@(t) cos(1 / 16 * pi * t), t);
 8   title('The Function g'); xlabel('t'); ylabel('g(t)');
 9   subplot(2, 1, 2);
10   fplot(@(t) sinc(t / pi), t);
11   title('The Function h'); xlabel('t'); ylabel('h(t)');
12   print('output.pdf', '-dpdf', '-bestfit');
```



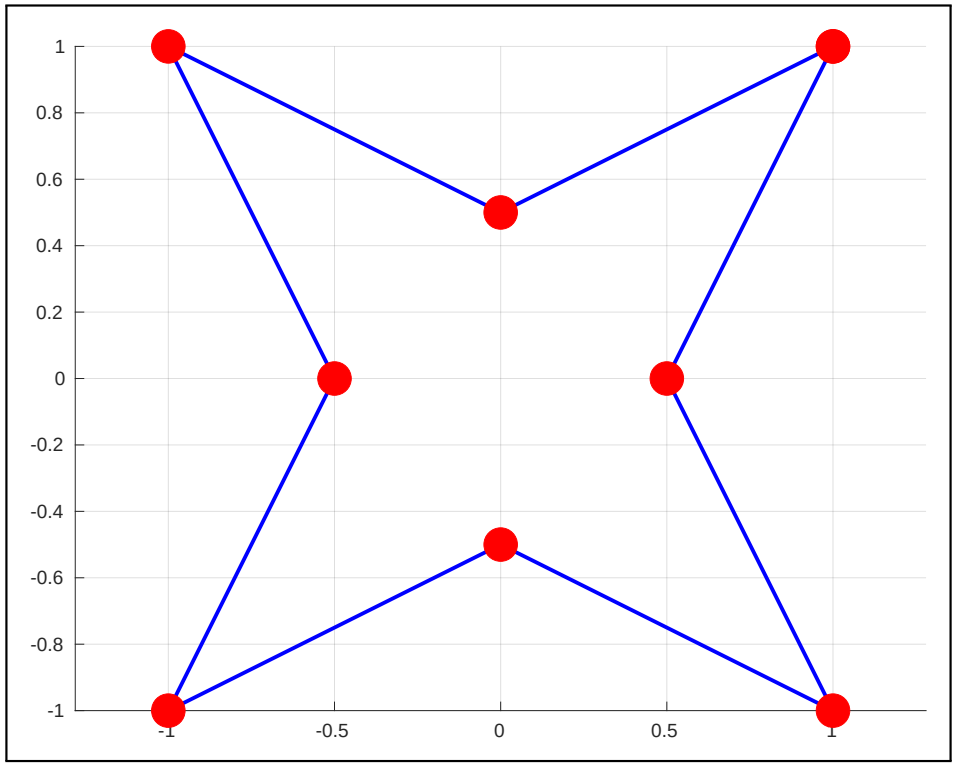
Example: Plot of Sequences [stem]

```
1   figure
2   n = 0 : 50;
3   hold on
4   stem(n, cos(3/32 * pi * n), 'filled', 'r', 'markersize', 5);
5   stem(n, 0.5*sin(1/16 * pi * n), 'filled', 'b', 'markersize', 5);
6   hold off
7   legend('cos(3{\pi}n/32)', 'sin({\pi}n/16)/2');
8   print('output.pdf', '-dpdf', '-bestfit', '-opengl');
```



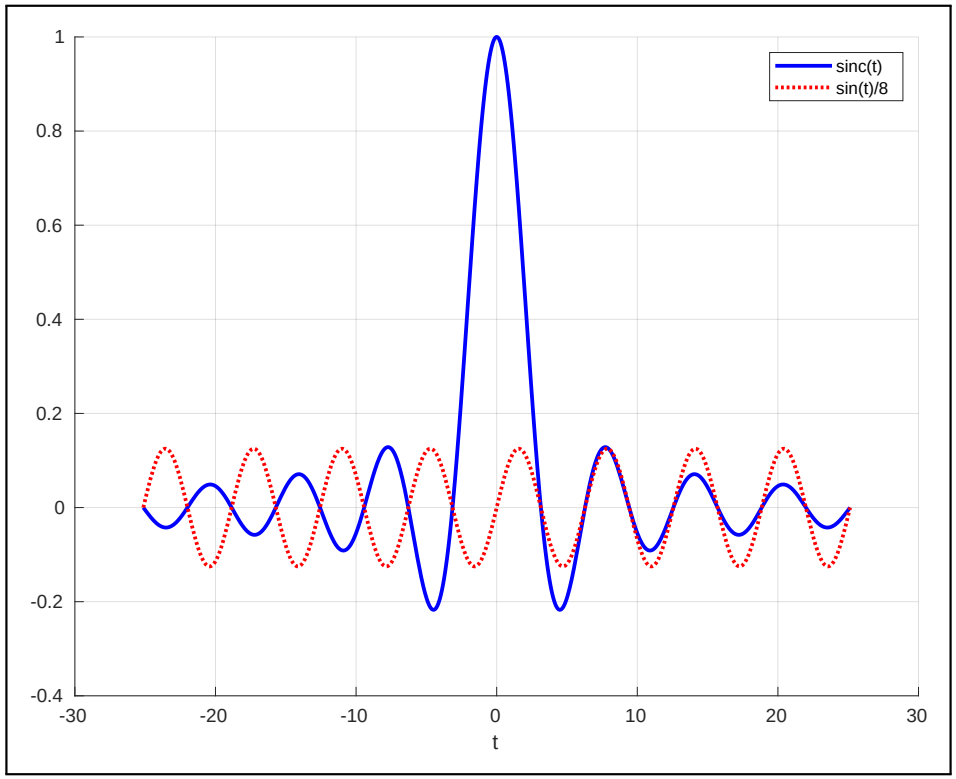
Example: Plot of Points/Path in the Plane [plot, axis]

```
1   figure
2   z = [1+j 0.5 1-j -0.5*j -1-j -0.5 -1+j 0.5*j 1+j];
3   hold on
4   plot(real(z), imag(z), 'b-', 'linewidth', 2);
5   plot(real(z), imag(z), 'r.', 'markersize', 60);
6   hold off
7   axis equal
8   grid on
9   print('output.pdf', '-dpdf', '-bestfit');
```



Example: Plotting Directly From Function [fplot]

```
1   figure
2   hold on
3   t = [-8 8] * pi;
4   fplot(@(t) sinc(t / pi), t, 'b-', 'linewidth', 2);
5   fplot(@(t) sin(t) / 8, t, 'r:', 'linewidth', 2);
6   hold off
7   grid on
8   xlabel('t');
9   legend('sinc(t)', 'sin(t)/8');
10  print('output.pdf', '-dpdf', '-bestfit');
```



Saving and Printing Figures

- The `print` function is provided for saving or printing copies of figures.
- MATLAB supports the generation of figures in many formats, including:
  - Portable Document Format (PDF)
  - PostScript (PS)
  - Encapsulated PostScript (EPS)
  - ISO/IEC 10918 (JPEG)
  - Portable Network Graphics (PNG)
  - Scalable Vector Graphics (SVG)
- When saving figures that contain lines and/or text, vector formats (such as PDF) should be preferred over raster formats (such as JPEG and PNG), as vector formats allow distortion-free scaling.
- In many of the earlier plotting examples, the figures generated were saved to a file in PDF format using the `print` function.
- The `print` function can also be used to send output directly to a printer.
- For more information on the `print` function, use the help browser (i.e., "`doc print`").

Additional Plotting Examples

- Several additional examples demonstrating various plotting capabilities of MATLAB can be found in the Examples section, including:
  - A plot of a univariate function.
  - A plot of a bivariate function as a shaded surface.
  - A plot of a bivariate function as a surface with contour lines.
  - A stem plot of a sequence.
  - A histogram.
  - A pie chart.

Section 15.11

Symbolic Computation

Symbolic Computation

- Symbolic computation is sometimes quite helpful in solving engineering problems.
- For example, a very complicated formula or equation involving several variables might need to be simplified without assuming specific values for the variables in the formula/equation.
- The Symbolic Math Toolbox provides MATLAB with such symbolic computation capabilities.
- Some computations that can be performed symbolically include:
  - simplification of formulas
  - integration
  - differentiation
  - solving systems of equations
  - transforms (e.g., forward/inverse unilateral Laplace transform)
  - plotting functions expressed in symbolic form

Symbolic Objects

- The Symbolic Math Toolbox defines a new data type called a **symbolic object**.
- The toolbox uses symbolic objects to represent symbolic variables, constants, and expressions.
- A symbolic object can have as its value any valid mathematical expression.
- Symbolic objects can be used in many of the same ways as non-symbolic objects.
- One must, however, keep in mind that performing a computation symbolically is quite different than performing it non-symbolically.
- Generally speaking, symbolic computation is much slower than non-symbolic computation.

Creating Symbolic Objects

- A symbolic variable is created using either the `sym` function or `syms` directive.
- For example, a symbolic variable `x` (whose value is simply itself) can be created using the `sym` function with the code:

```
x = sym('x');
```

- The same result can be achieved in a less verbose manner using the `syms` directive as follows:

```
syms x;
```

- The `syms` directive allows multiple variables to be specified.
- For example, symbolic variables named `x`, `y`, and `z` can be created with the code:

```
syms x y z;
```

- The `sym` function can also be used to create symbolic constants.
- For example, we can create a symbolic constant `p` that has the value π with the code:

```
p = sym(pi);
```

Creating Symbolic Objects (Continued)

- The character string argument passed to the `sym` function can only contain a variable name or constant (i.e., it cannot be an arbitrary expression).
- For example, code like the following is ***not allowed***:

```
f = sym('a * x ^ 2 + b * x + c');
  % ERROR: cannot pass arbitrary expression to
  % sym function
```

- From symbolic variables and constants, more complicated symbolic expressions can be constructed.
- For example, a symbolic expression `f` with the value `a * x ^ 2 + b * x + c` can be created with the code:

```
syms a b c x;
f = a * x ^ 2 + b * x + c;
```

Symbolic Versus Non-Symbolic Objects

- Symbolic objects can often be used in similar ways as non-symbolic objects.
- For example, we can do things like:

```
syms t;
f = t + 1;
g0 = f ^ 3 - 2 * f - 21;
g1 = cos(f) * sin(f / 3);
```

- Although symbolic and non-symbolic objects can often be used in similar ways, they are very different things.
- For example, the following two lines of code have very different effects:

```
x = pi;
x = sym(pi);
```

Substitution With Symbolic Objects

- To substitute some expression/variable for another variable, use the `subs` function.
- For example, to substitute `t + 1` for `t` in the expression `t ^ 2`, we can use the following code:

```
syms t;
f = t ^ 2;
g = subs(f, t, t + 1)
```

- After executing the preceding code, `g` is associated with the expression:

```
(t + 1) ^ 2
```

Factoring of Symbolic Objects

- To factor a symbolic expression, use the `factor` function.
- For example, suppose that we want to factor the polynomial $t^2 + 3t + 2$.
- This could be accomplished with the following code:

```
syms t;
f = t ^ 2 + 3 * t + 2;
g = factor(f)
```

- After executing the preceding code, `g` is associated with the (row-vector) expression:

```
[t + 2, t + 1]
```

- Note that, by default, the `factor` function will only produce factors with real roots.
- To obtain factors with complex roots, the `'FactorMode'` option must be used; for example:

```
syms z;
g = factor(z ^ 2 + 4, 'FactorMode', 'complex')
```

Simplification of Symbolic Objects

- To simplify a symbolic expression, use the `simplify` function.
- For example, suppose that we want to substitute `2 * t + 1` for `t` in the expression `t ^ 2 - 1` and simplify the result.
- This can be accomplished with the following code:

```
syms t;
f = t ^ 2 - 1;
g = simplify(subs(f, t, 2 * t + 1))
```

- After executing the preceding code, `g` is associated with the expression

```
(2 * t + 1) ^ 2 - 1
```

Expansion of Symbolic Objects

- To expand an expression, use the `expand` function.
- For example, to compute $(t+1)^5$, we can use the following code:

```
syms t;
f = (t + 1) ^ 5;
g = expand(f)
```

- After executing the preceding code, `g` is associated with the expression:

```
t ^ 5 + 5 * t ^ 4 + 10 * t ^ 3 + 10 * t ^ 2 + 5 * t + 1
```

Pretty-Printing of Symbolic Objects

- To display an expression in a human-friendly format, use the `pretty` function.
- For example, to compute $\left[\frac{1}{2}t^2 + \frac{1}{3}(t+1)\right]^{-4}$ in an expanded and beautified format, we can use the following code:

```
syms t;
f = ((1/2) * t^2 + (1/3) * (t+1))^(-4);
pretty(expand(f))
```

- The output of the `pretty` function in this case might look something like:

```
                                                1
-------------------------------------------------------------------------
 8     7     6         5         4         3        2
t     t     t     11 t     65 t     22 t     4 t     4 t     1
-- + -- + -- + ----- + ----- + ----- + ---- + --- + --
16    6     3     27       162       81       27       81    81
```

Plotting Symbolic Expressions

- To plot a symbolic expression in one variable, the `ezplot` function can be used.
- For example, to plot the function $f(t) = 3t^2 - 4t + 2$, we can use the code:

```
syms t;
ezplot(3 * t ^ 2 - 4 * t + 2);
```

- The range of the independent variable may optionally be specified.
- For example, to plot the function $f(t) = 3t^2 - 4t + 2$ over the interval $[-1, 1]$, we can use the code:

```
syms t;
ezplot(3 * t ^ 2 - 4 * t + 2, [-1 1]);
```

Simplification Example

sum_of_arithmetic_sequence.m

```
% Compute the formula for the sum of the arithmetic
% sequence:
%    a, a + d, a + 2d, ... a + (n - 1)d.
clear
syms a d k n
f = simplify(symsum(a + k * d, k, 0, n - 1));
pretty(f);
```

Command Window Output

```
>> sum_of_arithmetic_sequence
      d n (n - 1)
a n + -----------
          2

>>
```

Integration Example

integration_1.m

```
% Compute the integral, over the interval [a, b], of the
% function:
%    x(t) = t sin(t) e^(-t)
clear
syms t a b
f = int(t * sin(t) * exp(-t), a, b);
pretty(f);
```

Command Window Output

```
>> integration_1
exp(-a) (cos(a) + a cos(a) + a sin(a))
-------------------------------------
                  2

    exp(-b) (cos(b) + b cos(b) + b sin(b))
  - --------------------------------------
                     2

>>
```

Differentiation Example

differentiation_1.m

```
% Compute the derivative of the function:
%    x(t) = t cos(t) e^(-t)
clear
syms t
f = diff(t * cos(t) * exp(-t), t);
pretty(f);
```

Command Window Output

```
>> differentiation_1
exp(-t) cos(t) - t exp(-t) cos(t) - t exp(-t) sin(t)

>>
```

Unilateral Laplace Transform Example

ult_1.m

```
% Compute the (forward) unilateral Laplace transform of
% the function:
%    x(t) = t e^(-t) sin(t)
clear
syms t
f = laplace(t * exp(-t) * sin(t));
pretty(f);
```

Command Window Output

```
>> ult_1
    2 s + 2
---------------
        2     2
((s + 1)  + 1)

>>
```

Inverse Unilateral Laplace Transform Example

iult_1.m

```
% Compute the inverse unilateral Laplace transform of
% the function:
%    X(s) = 1 / [(s + 1) (s + 2)^2]
clear
syms s
f = ilaplace(1 / ((s + 1) * (s + 2)^2));
pretty(f);
```

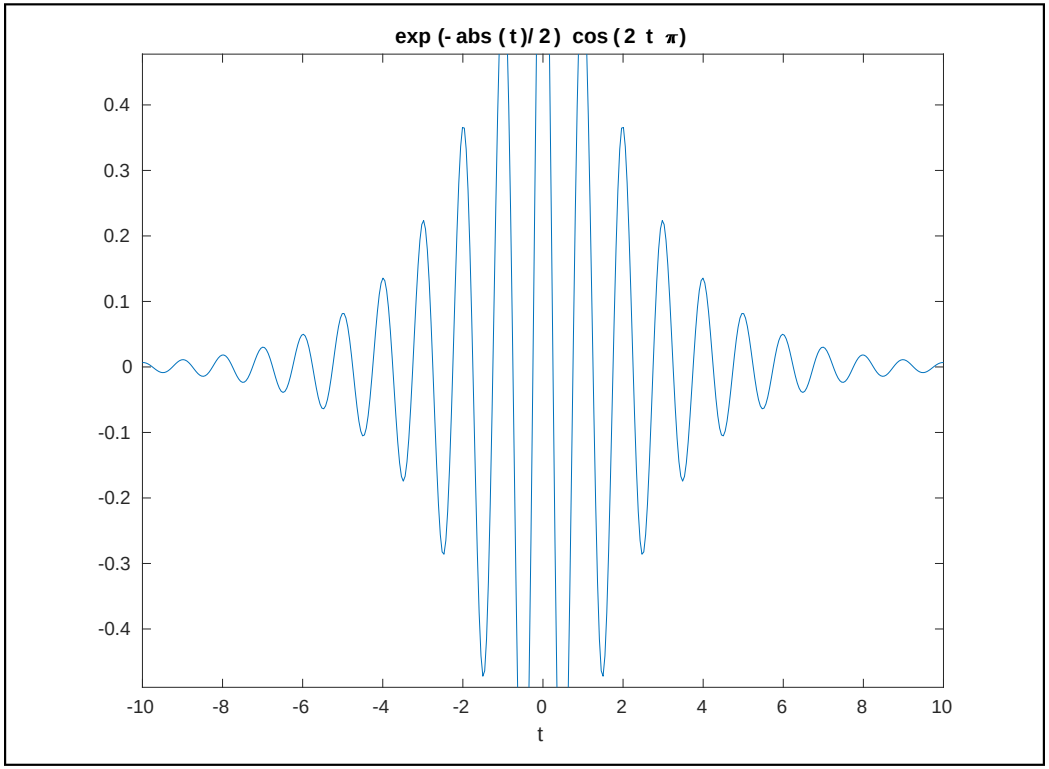
Command Window Output

```
>> iult_1
exp(-t) - exp(-2 t) - t exp(-2 t)

>>
```

Plotting Example

```
1  clear
2  syms t
3  ezplot(cos(2 * pi * t) * exp(-abs(t)/2), [-10 10]);
4  print('output.pdf', '-dpdf', '-bestfit');
```



Section 15.12

Continuous-Time Signal Processing

Transfer Functions

- Most CT LTI systems of practical interest are causal with rational transfer functions.
- For this reason, MATLAB has considerable functionality for working with such systems.
- Consider the rational transfer function H with a RHP ROC, where

$$H(s) = \frac{\sum_{k=1}^{m} a_k s^{m-k}}{\sum_{k=1}^{n} b_k s^{n-k}} = \frac{a_1 s^{m-1} + a_2 s^{m-2} + \ldots + a_m}{b_1 s^{n-1} + b_2 s^{n-2} + \ldots + b_n}.$$

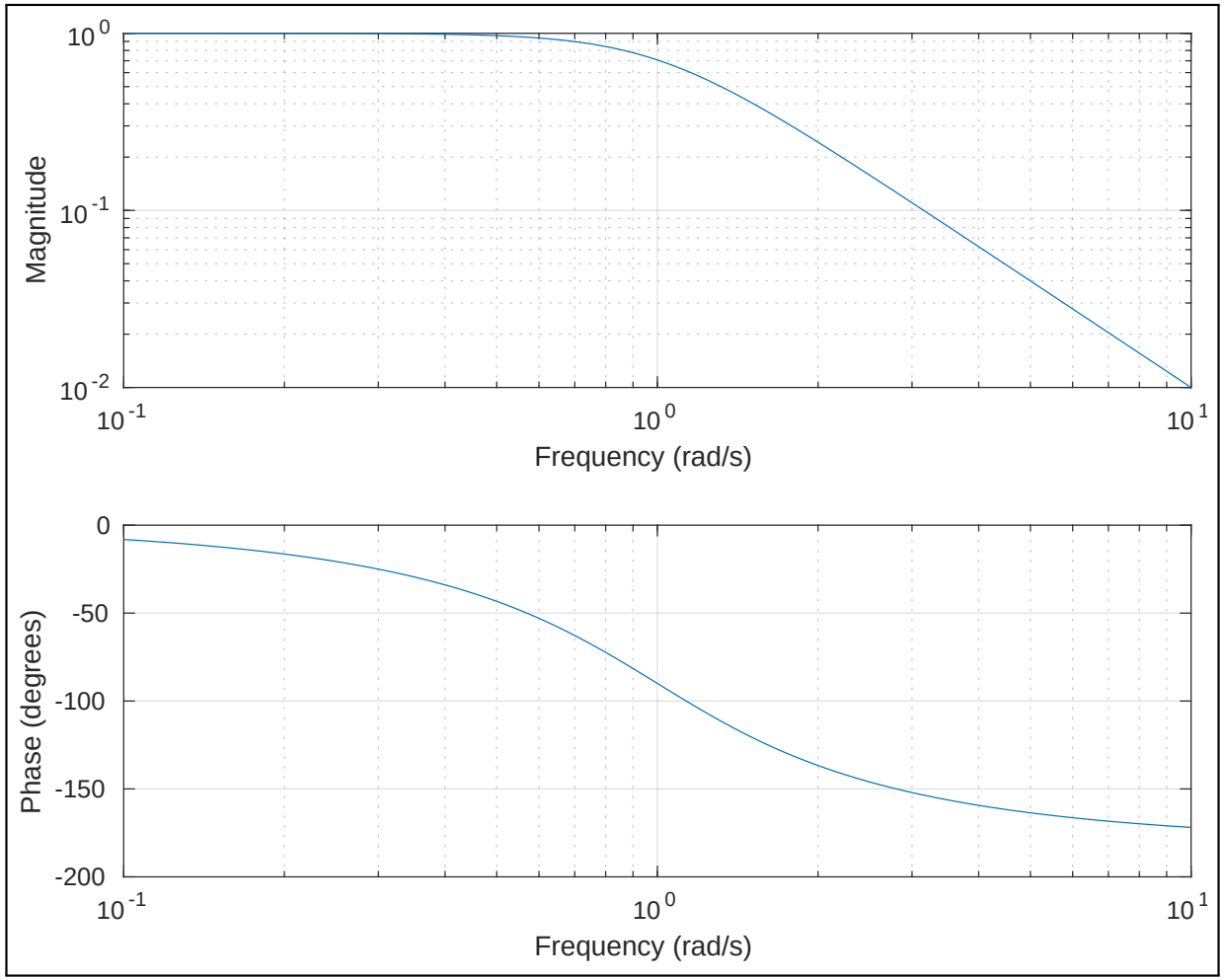
- Typically, MATLAB represents such a transfer function using two vectors of coefficients, one for the $\{a_k\}$ and one for the $\{b_k\}$.
- For example, $H(s) = \frac{s-1}{s^2+3s+2}$ would be represented by the vectors `[1 -1]` and `[1 3 2]`.

Frequency Response

- Consider a causal LTI (CT) system with a transfer function H of the form $H(s) = \frac{\sum_{k=1}^{m} a_k s^{m-k}}{\sum_{k=1}^{n} b_k s^{n-k}} = \frac{a_1 s^{m-1} + a_2 s^{m-2} + \ldots + a_m}{b_1 s^{n-1} + b_2 s^{n-2} + \ldots + b_n}$.
- The frequency response of such a system can be computed and optionally plotted using the `freqs` function.
- The `freqs` function has the interface:
  - `[freq_resp w] = freqs(a, b, w)`
  - `a` is vector whose elements are $a_1, a_2, \ldots, a_m$
  - `b` is vector whose elements are $b_1, b_2, \ldots, b_n$
  - `w` is a vector whose elements are the frequencies at which to calculate the frequency response (and may be omitted in which case `w` is chosen by the `freqs` function)
  - `freq_resp` is the vector of frequency response values
- If the return value of the `freqs` function is discarded, the function also plots magnitude and phase responses, with the magnitude and frequency plotted on logarithmic scales.

Example: Plotting Frequency Response With `freqs`

```
1  freqs([1], [1 sqrt(2) 1]);
2  print('output.pdf', '-dpdf', '-bestfit');
```

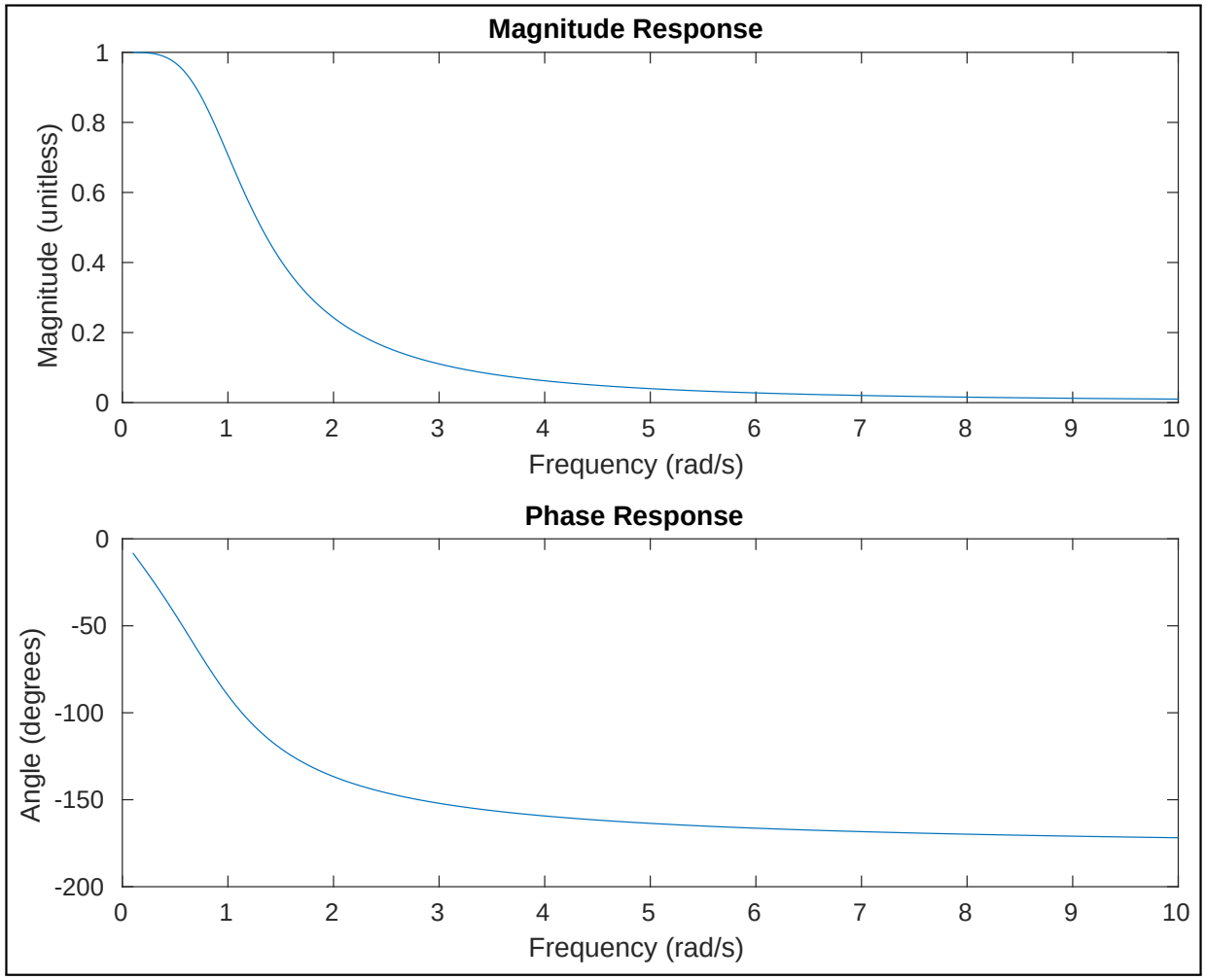


Example: Plotting Frequency Response With Linear Scale

```
1  function [freq_resp, omega] = my_freqs(tf_num, tf_denom, omega)
2      % my_freqs - Compute and optionally plot the frequency response.
3      % The optional plot uses a linear scale for both axes.
4      if nargin >= 3
5          [freq_resp, omega] = freqs(tf_num, tf_denom, omega);
6      else
7          [freq_resp, omega] = freqs(tf_num, tf_denom);
8      end
9      if nargout == 0
10         mag_resp = abs(freq_resp);
11         phase_resp = angle(freq_resp) / pi * 180;
12         subplot(2, 1, 1);
13         plot(omega, mag_resp);
14         title('Magnitude Response');
15         xlabel('Frequency (rad/s)');
16         ylabel('Magnitude (unitless)');
17         subplot(2, 1, 2);
18         plot(omega, phase_resp);
19         title('Phase Response');
20         xlabel('Frequency (rad/s)');
21         ylabel('Angle (degrees)');
22     end
23 end
```

Example: Plotting Frequency Response With `my_freqs`

```
1  my_freqs([1], [1 sqrt(2) 1]);
2  print('output.pdf', '-dpdf', '-bestfit');
```

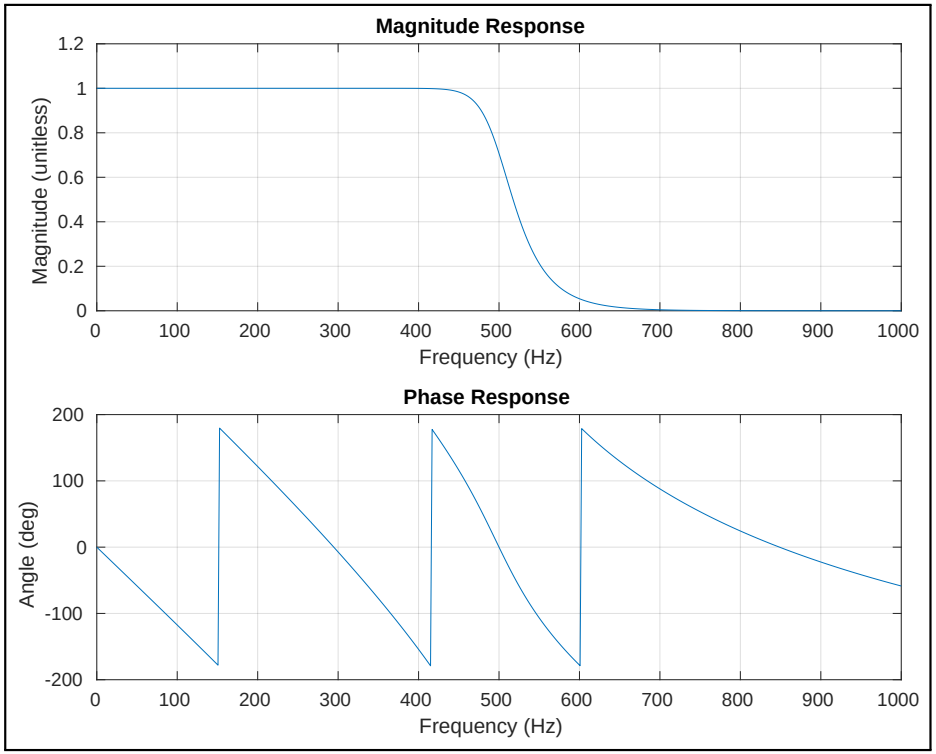


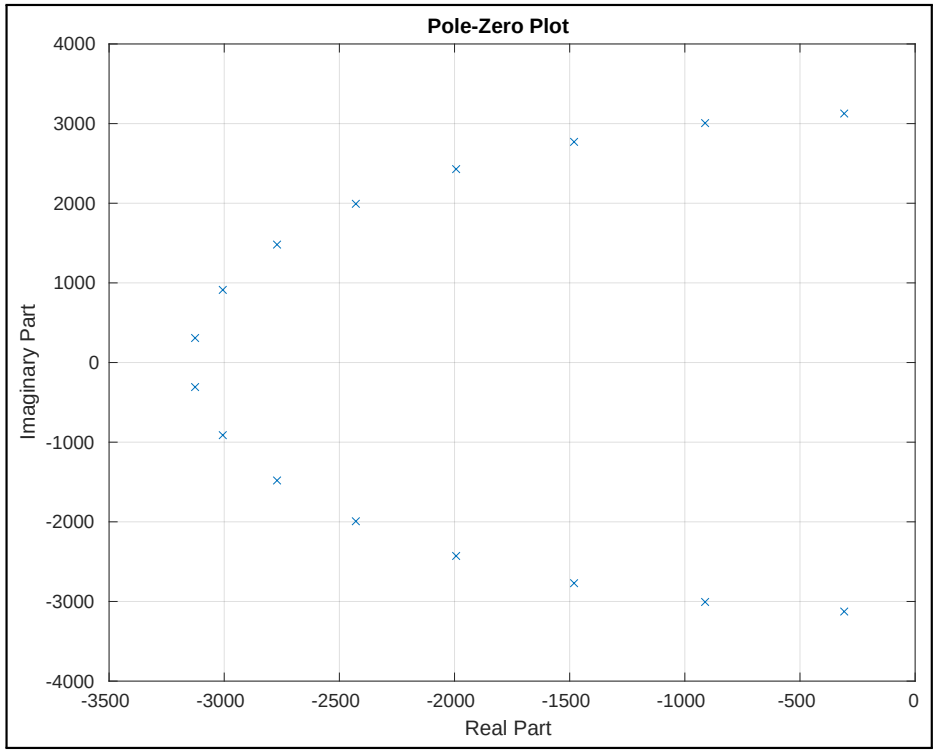
Functions for Continuous-Time Filter Design

Name	Description
`besself`	design (CT) Bessel filter
`butter`	design (CT) Butterworth filter
`cheby1`	design (CT/DT) Chebyshev Type 1 filter
`cheby2`	design (CT/DT) Chebyshev Type 2 filter
`ellip`	design (CT/DT) elliptic or Cauer filter

Butterworth Lowpass Filter Design

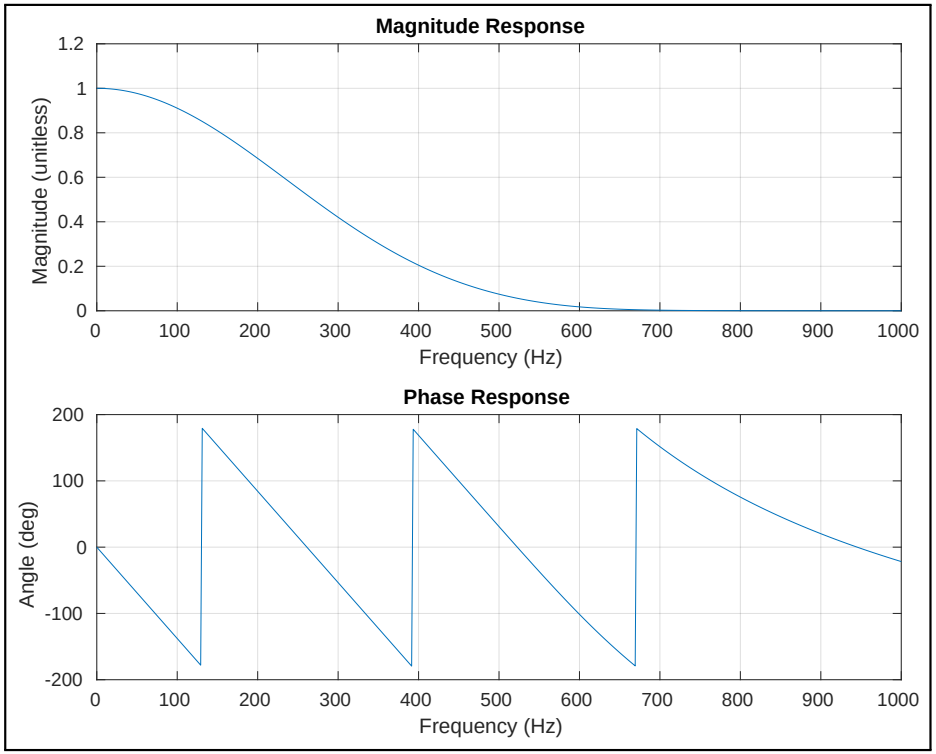
```
% 16th-order Butterworth (CT) lowpass filter with
% cutoff frequency 500 Hz
[a b] = butter(16, 2 * pi * 500, 'low', 's');
```

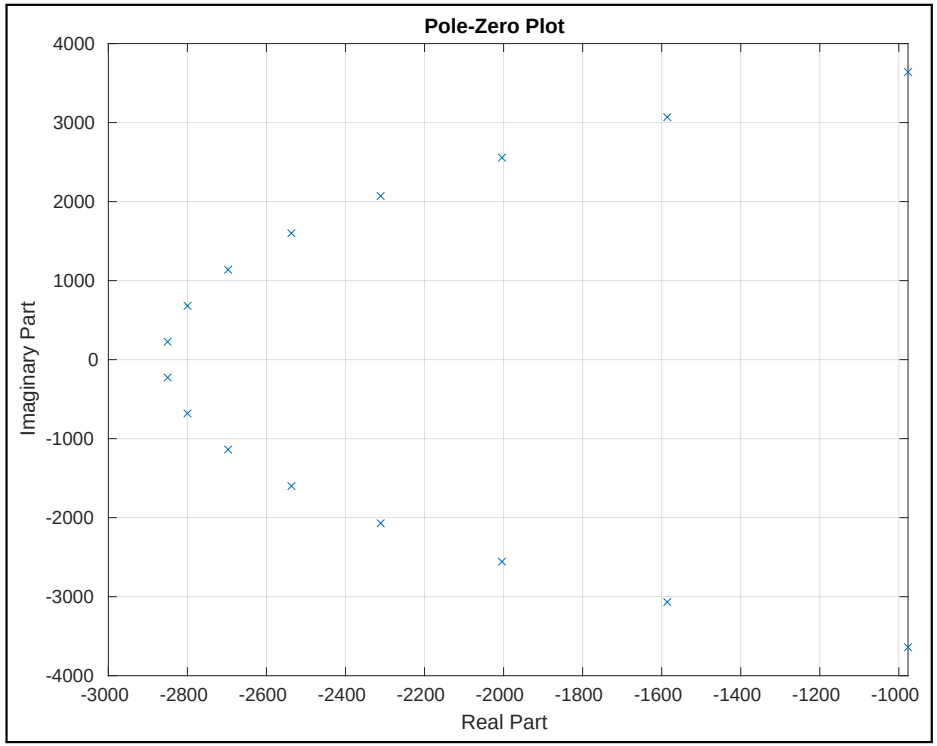




Bessel Lowpass Filter Design

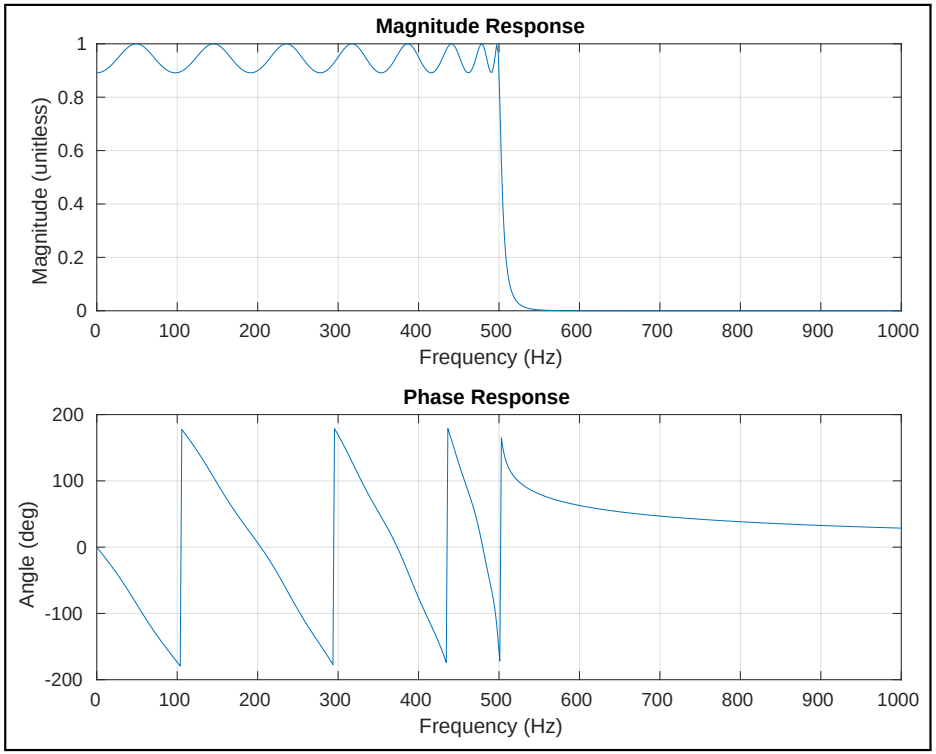
```
% 16th-order Bessel (CT) lowpass filter with
% cutoff frequency 500 Hz
[a b] = besself(16, 2 * pi * 500);
```

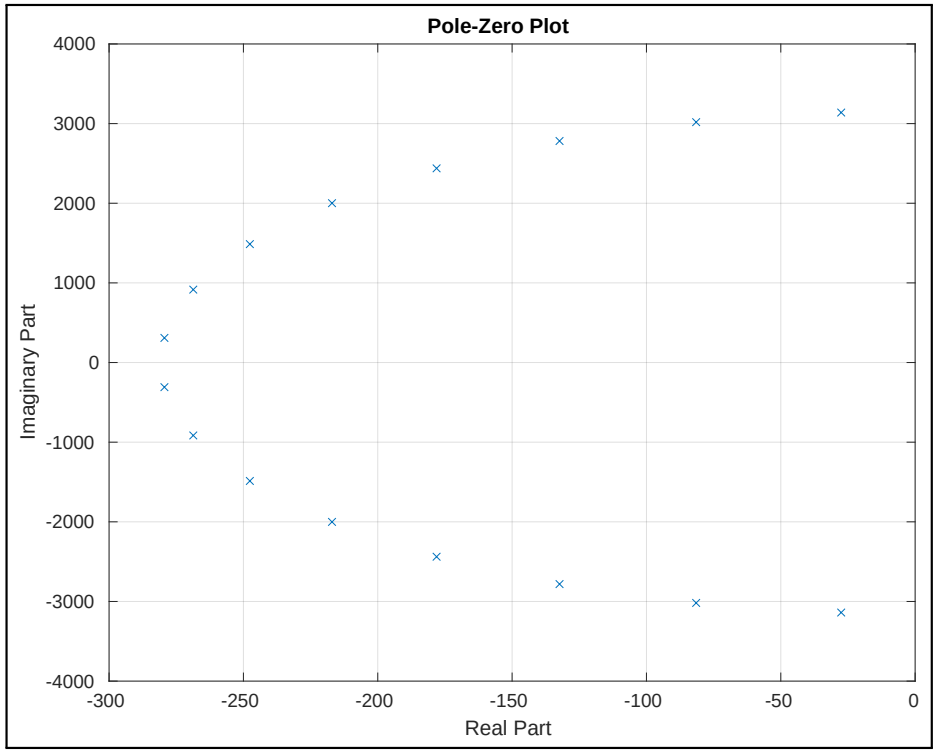




Chebyshev Type-1 Lowpass Filter Design

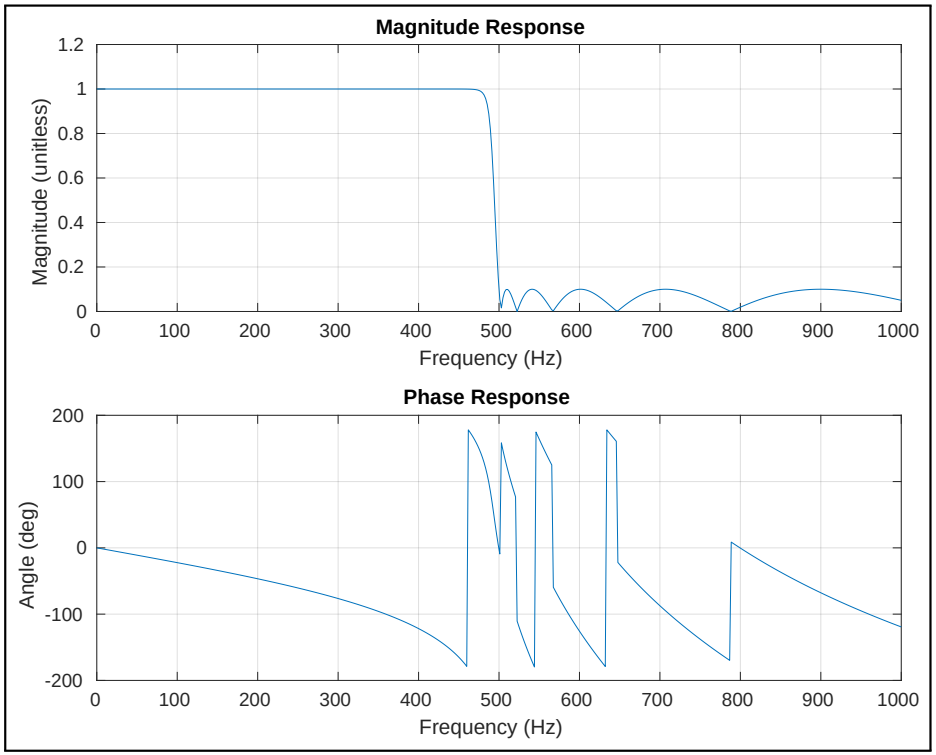
```
% 16th-order Chebyshev Type-1 (CT) lowpass filter with
% cutoff frequency 500 Hz and 1 dB passband ripple
[a b] = cheby1(16, 1, 2 * pi * 500, 'low', 's');
```

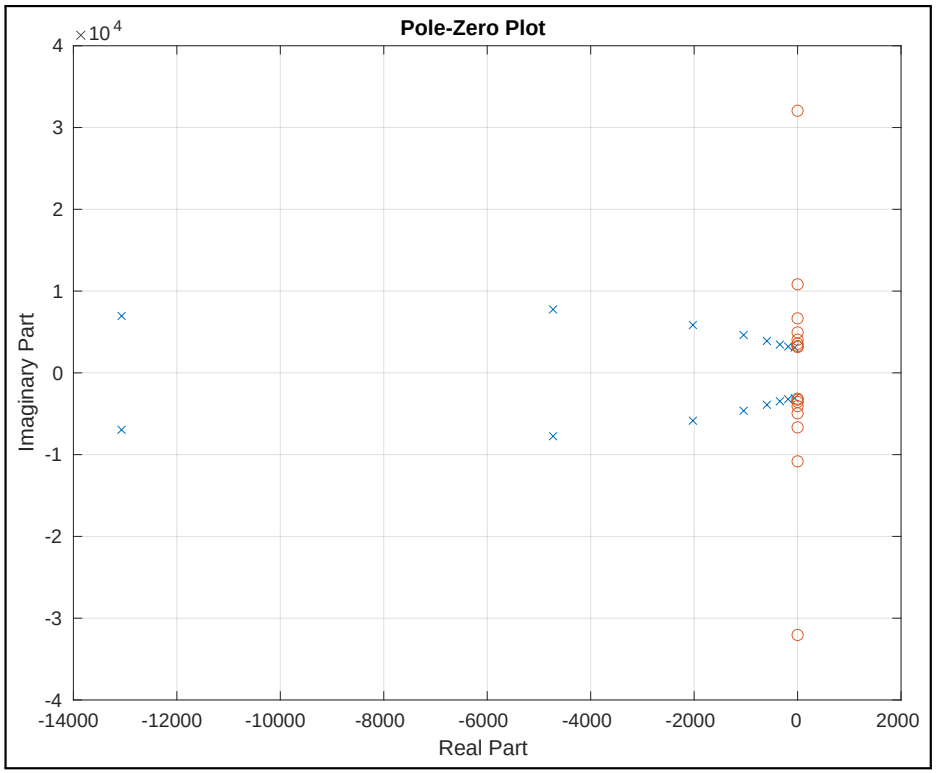




Chebyshev Type-2 Lowpass Filter Design

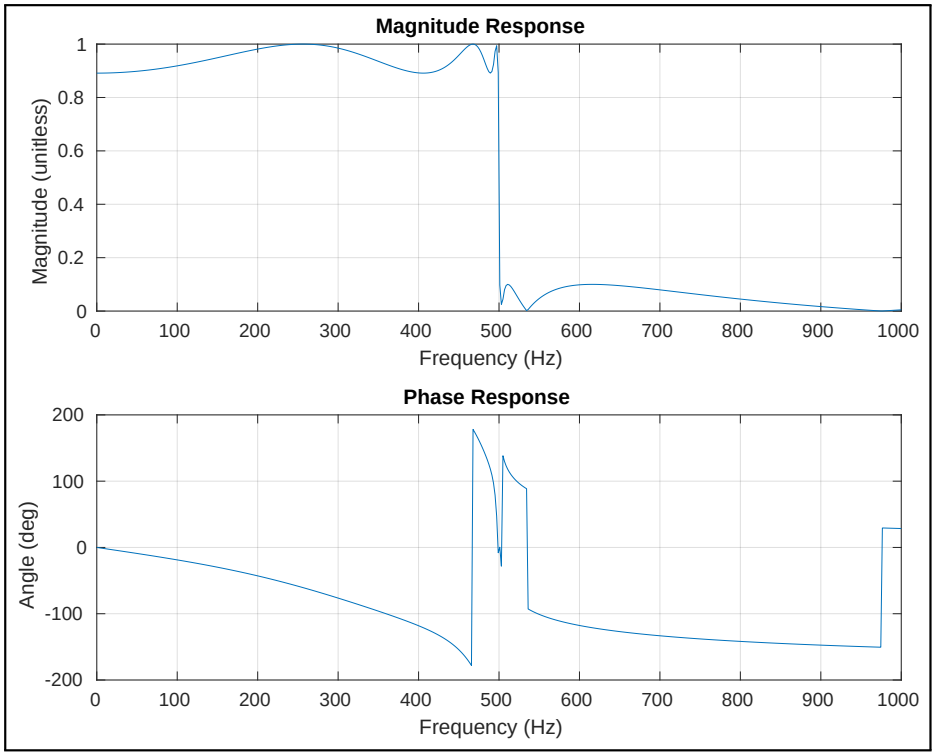
```
% 16th-order Chebyshev Type-2 (CT) lowpass filter with
% cutoff frequency 500 Hz and 20 dB stopband ripple
[a b] = cheby2(16, 20, 2 * pi * 500, 'low', 's');
```

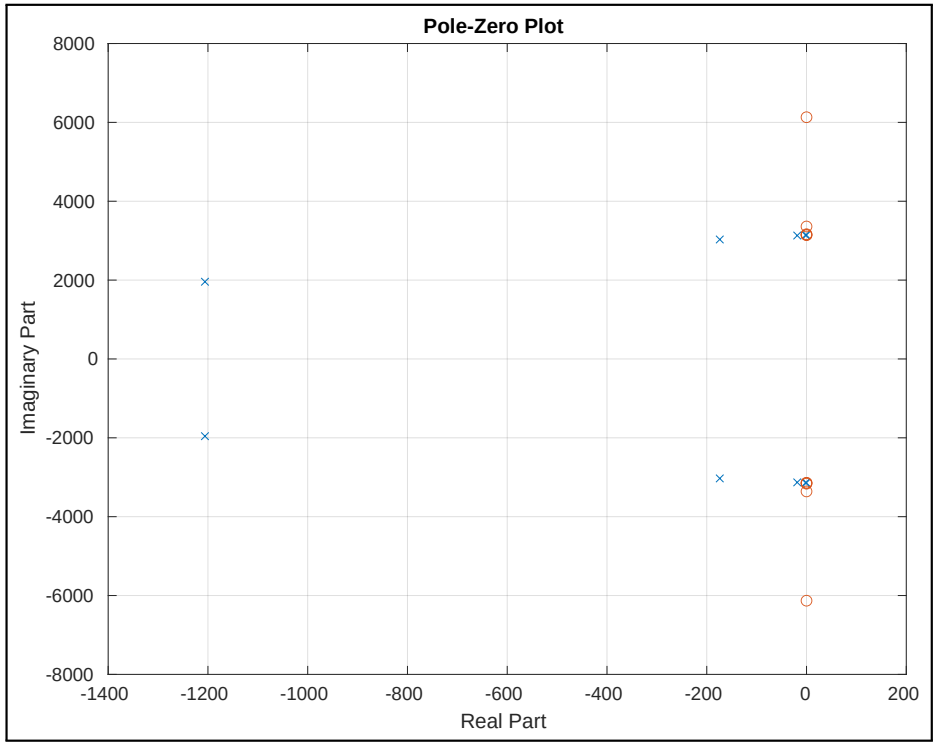




Elliptic Lowpass Filter Design

```
% 16th-order elliptic (CT) lowpass filter with
% cutoff frequency 500 Hz, 1 dB passband ripple, and
% 20 dB stopband ripple
[a b] = ellip(10, 1, 20, 2 * pi * 500, 'low', 's');
```





Section 15.13

Discrete-Time Signal Processing

Transfer Functions

- Most DT LTI systems of practical interest are causal with rational transfer functions.
- For this reason, MATLAB has considerable functionality for working with such systems.
- Consider the rational transfer function

$$H(z) = \frac{\sum_{k=1}^{m} a_k z^{-(k-1)}}{\sum_{k=1}^{n} b_k z^{-(k-1)}} = \frac{a_1 + a_2 z^{-1} + \ldots + a_m z^{-(m-1)}}{b_1 + b_2 z^{-1} + \ldots + b_n z^{-(n-1)}}.$$

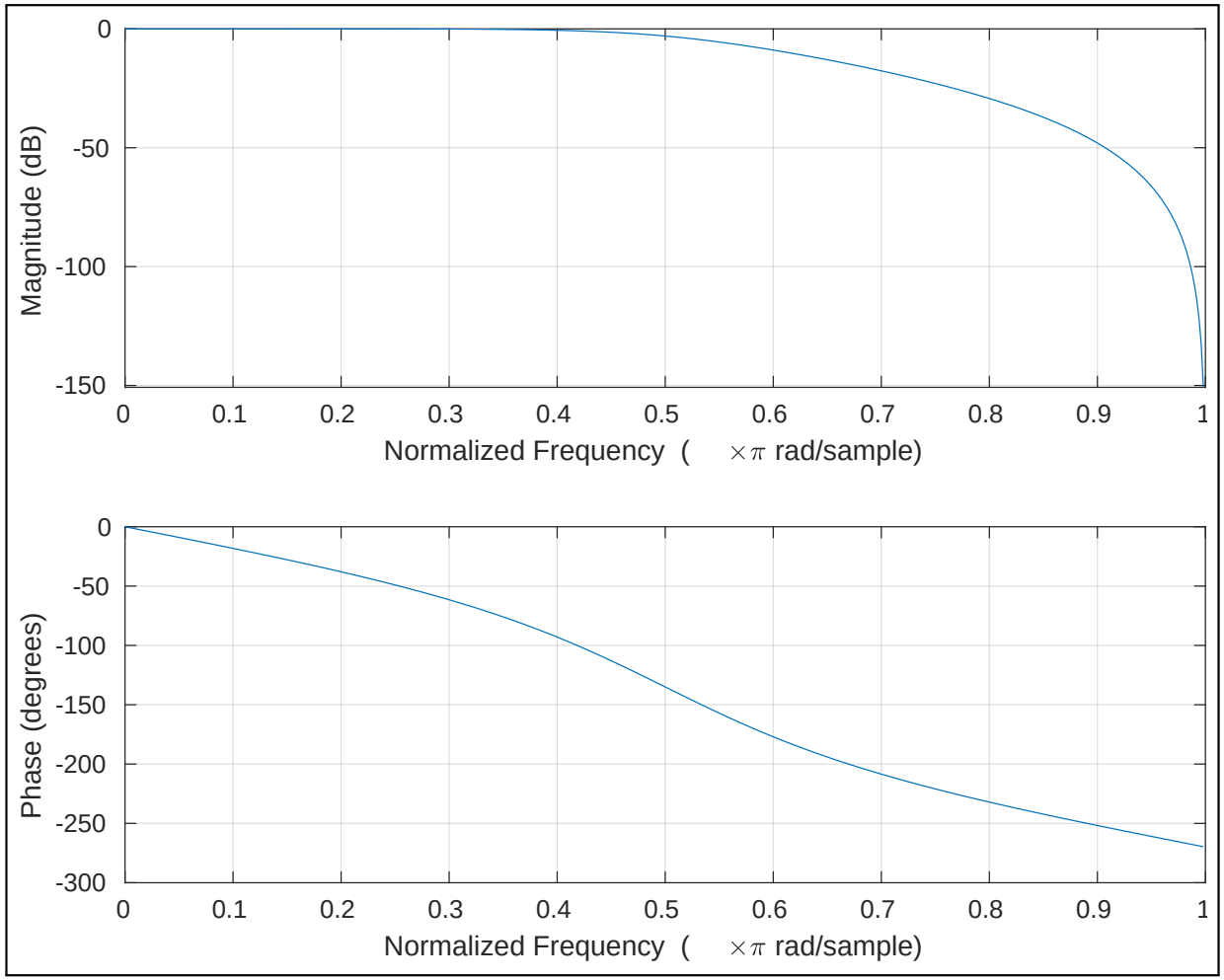
- Typically, MATLAB represents such a transfer function using two vectors of coefficients, one for the $\{a_k\}$ and one for the $\{b_k\}$.
- For example, $H(z) = \frac{1+z^{-1}}{6+5z^{-1}+z^{-2}}$ would be represented by the vectors `[1 1]` and `[6 5 1]`.

Frequency Response

- Consider a causal LTI (DT) system with a transfer function H of the form $H(z) = \frac{\sum_{k=1}^{m} a_k z^{-(k-1)}}{\sum_{k=1}^{n} b_k z^{-(k-1)}} = \frac{a_1 + a_2 z^{-1} + \ldots + a_m z^{-(m-1)}}{b_1 + b_2 z^{-1} + \ldots + b_n z^{-(n-1)}}$.
- The frequency response of such a system can be computed and optionally plotted using the `freqz` function.
- The `freqz` function has the interface:
  - `[freq_resp w] = freqz(a, b, w)`
  - `a` is vector whose elements are $a_1, a_2, \ldots, a_m$
  - `b` is vector whose elements are $b_1, b_2, \ldots, b_n$
  - `w` is a vector whose elements are the frequencies at which to calculate the frequency response (and may be omitted in which case `w` is chosen by the `freqz` function)
  - `freq_resp` is the vector of frequency response values
- If the return value of the `freqz` function is discarded, the function also plots magnitude and phase responses.

Example: Plotting Frequency Response With `freqz`

```
1  freqz([1/6 1/2 1/2 1/6], [1 0 1/3 0]);
2  print('output.pdf', '-dpdf', '-bestfit');
```

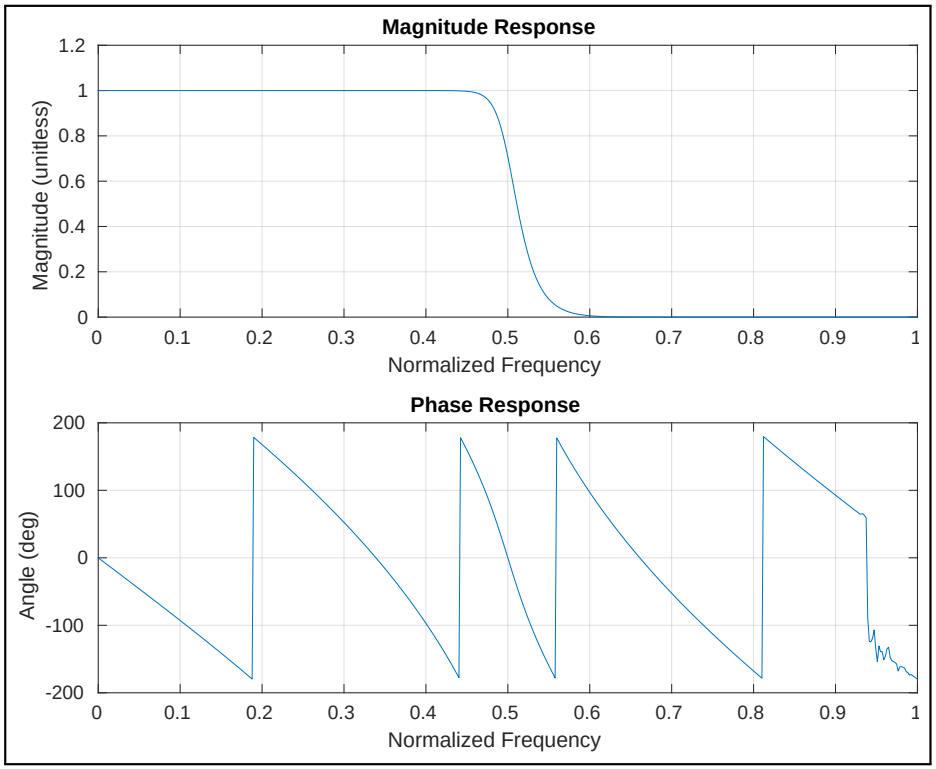


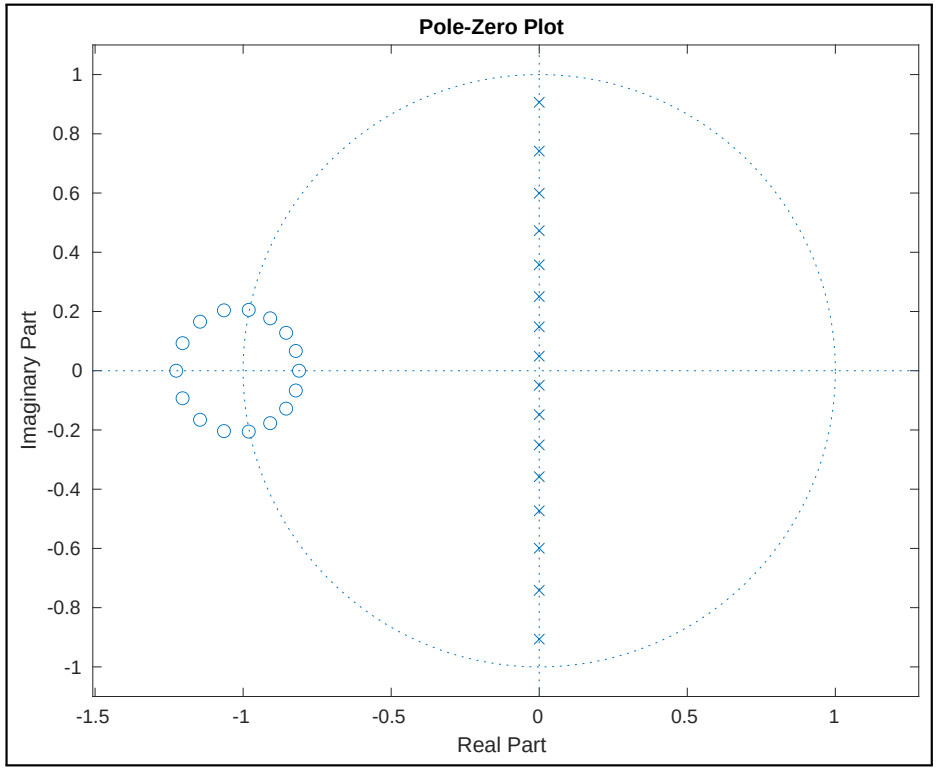
Functions for Discrete-Time Filter Design

Name	Description
`cheby1`	design (CT/DT) Chebyshev Type 1 filter
`cheby2`	design (CT/DT) Chebyshev Type 2 filter
`ellip`	design (CT/DT) elliptic or Cauer filter
`fir1`	design (DT) FIR filter using window method
`fir2`	design (DT) FIR filter using frequency-sampling method
`cfirpm`	complex and nonlinear-phase equiripple FIR filter design
`fircls`	linear-phase FIR filter design using constrained least-squares method
`firls`	linear-phase FIR filter design using least-squares error minimization
`firpm`	Parks-McClellan optimal equiripple FIR filter design

Butterworth Lowpass Filter Design

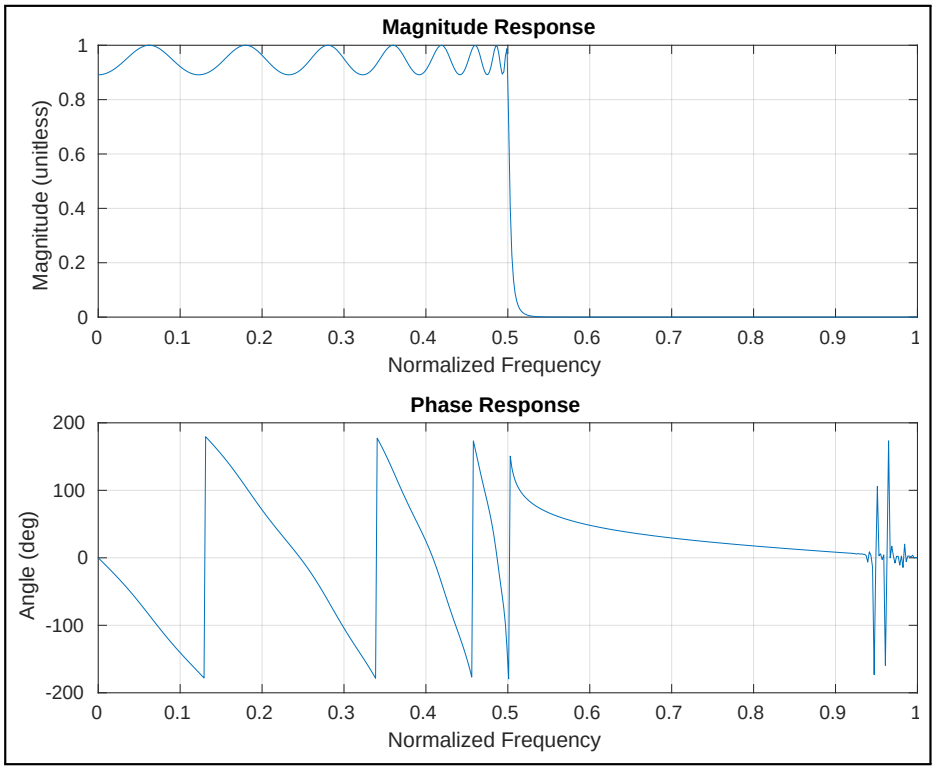
```
% 16th-order Butterworth (DT) lowpass filter with
% normalized cutoff frequency 0.5
[a b] = butter(16, 0.5, 'low');
```

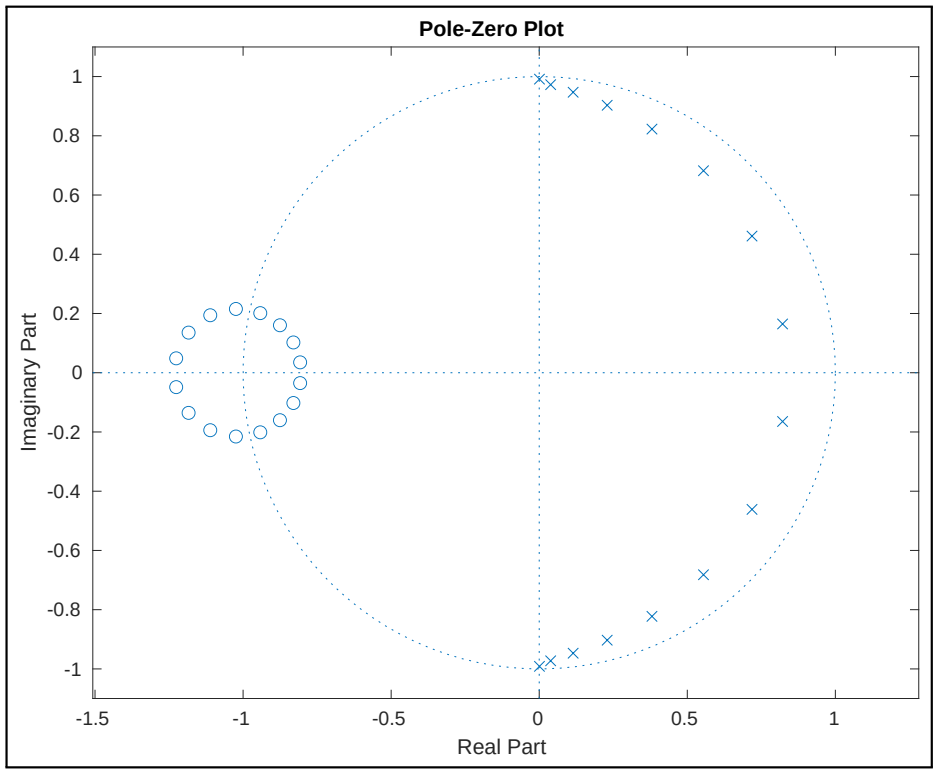




Chebyshev Type-1 Lowpass Filter Design

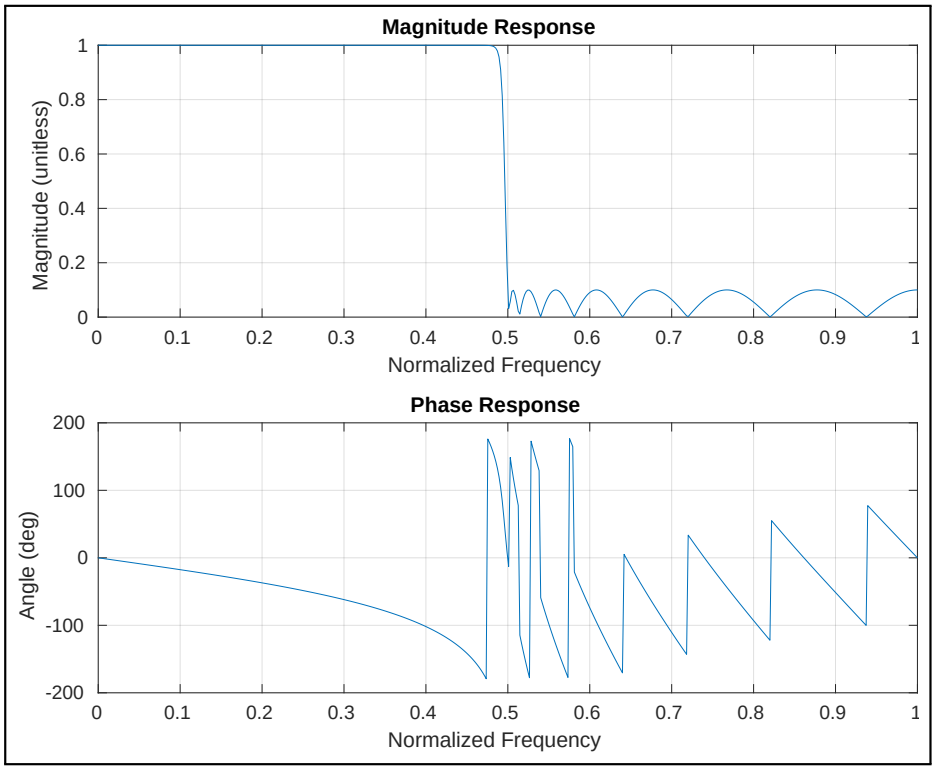
```
% 16th-order Chebyshev Type-1 (DT) lowpass filter with
% normalized cutoff frequency 0.5 and 1 dB passband ripple
[a b] = cheby1(16, 1, 0.5, 'low');
```

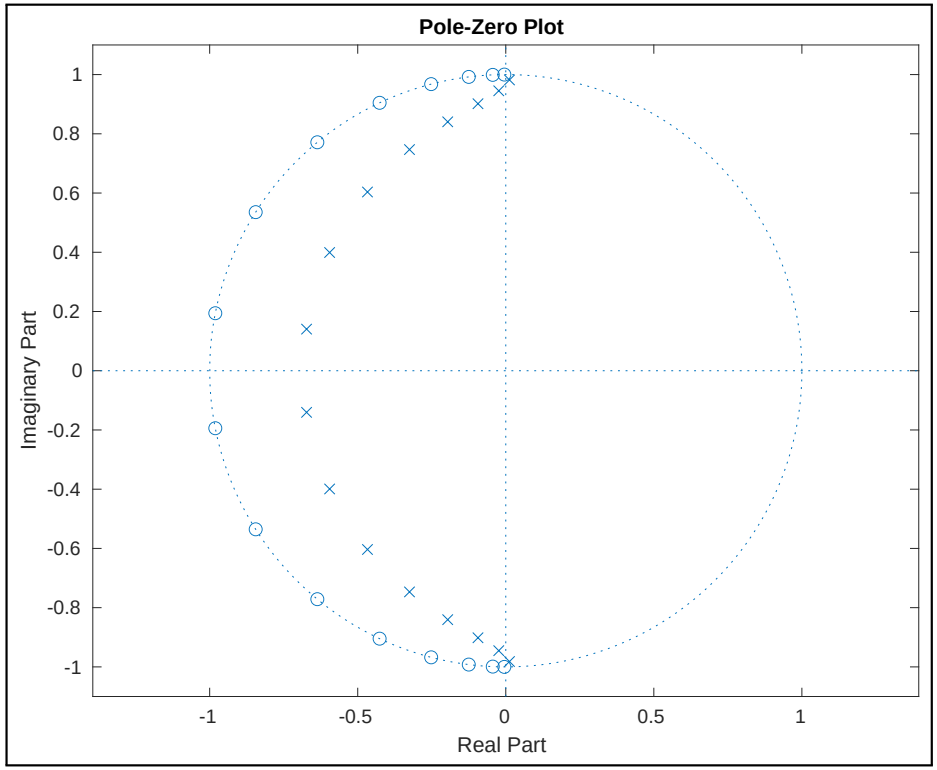




Chebyshev Type-2 Lowpass Filter Design

```
% 16th-order Chebyshev Type-2 (DT) lowpass filter with
% normalized cutoff frequency 0.5 and 20 dB stopband ripple
[a b] = cheby2(16, 20, 0.5, 'low');
```





Section 15.14

References

References I

1. B. D. Hahn and D. T. Valentine. *Essential MATLAB for Engineers and Scientists*. Butterworth-Heinemann, Burlington, MA, USA, 3rd edition, 2007.
2. S. J. Chapman. *MATLAB Programming for Engineers*. Cengage Learning, Boston, MA, USA, 5th edition, 2016.
3. S. Attaway. *MATLAB — A Practical Introduction to Programming and Problem Solving*. Butterworth-Heinemann, Cambridge, MA, USA, 5th edition, 2019.
4. H. Moore. *MATLAB for Engineers*. Pearson, Boston, MA, USA, 3rd edition, 2012.
5. A. Gilat. *MATLAB — An Introduction with Applications*. John Wiley & Sons, Hoboken, NJ, USA, 6th edition, 2017.
6. J. Sizemore and J. P. Mueller. *MATLAB for Dummies*. John Wiley & Sons, Hoboken, NJ, USA, 2015.

References II

7 D. M. Etter, D. C. Kuncicky, and D. Hull. *Introduction to MATLAB 6*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2004.

8 D. Hanselman and B. Littlefield. *Mastering MATLAB 6: A Comprehensive Tutorial and Reference*. Prentice Hall, Upper Saddle River, NJ, USA, 2001.

Part 16

Miscellany

Sum of Arithmetic and Geometric Sequences

- The sum of the arithmetic sequence $a, a+d, a+2d, \ldots, a+(n-1)d$ is given by

$$\sum_{k=0}^{n-1} (a+kd) = \frac{n[2a+d(n-1)]}{2}.$$

- The sum of the geometric sequence $a, ra, r^2a, \ldots, r^{n-1}a$ is given by

$$\sum_{k=0}^{n-1} r^k a = a\frac{r^n-1}{r-1} \quad \text{for } r \neq 1.$$

- The sum of the infinite geometric sequence $a, ra, r^2a, \ldots$ is given by

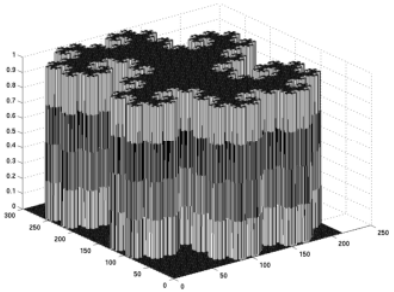
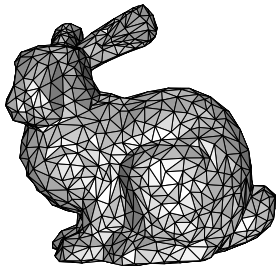
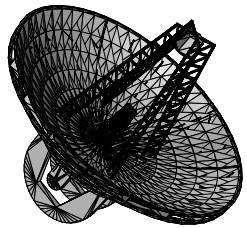
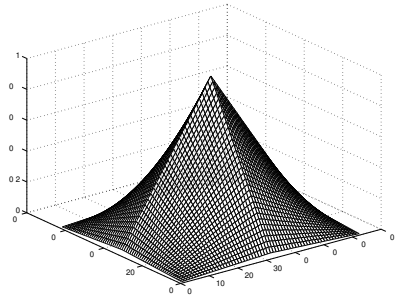
$$\sum_{k=0}^{\infty} r^k a = \frac{a}{1-r} \quad \text{for } |r| < 1.$$

Part 17

Epilogue

Other Courses Offered by the Author of These Lecture Slides

- If you did not suffer permanent emotional scarring as a result of using these lecture slides and you happen to be a student at the University of Victoria, you might wish to consider taking another one of the courses developed by the author of these lecture slides:
  - ***ECE 486: Multiresolution Signal and Geometry Processing with C++***
  - ***SENG 475: Advanced Programming Techniques for Robust Efficient Computing***
- For further information about the above courses (including the URLs for web sites of these courses), please refer to the slides that follow.

ECE 486/586: Multiresolution Signal and Geometry Processing with C++

- normally offered in Summer (May-August) term; only prerequisite ECE 310
- subdivision surfaces and subdivision wavelets
  - 3D computer graphics, animation, gaming (Toy Story, Blender software)
  - geometric modelling, visualization, computer-aided design
- multirate signal processing and wavelet systems
  - sampling rate conversion (audio processing, video transcoding)
  - signal compression (JPEG 2000, FBI fingerprint compression)
  - communication systems (transmultiplexers for CDMA, FDMA, TDMA)
- C++ (classes, templates, standard library), OpenGL, GLUT, CGAL
- software applications (using C++)
- for more information, visit course web page:
  `http://www.ece.uvic.ca/~mdadams/courses/wavelets`

SENG 475: Advanced Programming Techniques for Robust Efficient Computing (With C++)

- advanced programming techniques for robust efficient computing explored in context of C++ programming language
- topics covered may include:
  - concurrency, multithreading, transactional memory, parallelism, vectorization; cache-efficient coding; compile-time versus run-time computation; compile-time versus run-time polymorphism; generic programming techniques; resource/memory management; copy and move semantics; exception-safe coding
- applications areas considered may include:
  - geometry processing, computer graphics, signal processing, and numerical analysis
- open to any student with necessary prerequisites, which are:
  - SENG 265 or CENG 255 or CSC 230 or CSC 349A or ECE 255 or permission of Department
- for more information, see course web site:

  `http://www.ece.uvic.ca/~mdadams/courses/cpp`

Part 18

References

Online Resources I

1. Michael Adams. 2020-05 ECE 260 Video Lectures Playlist on YouTube. https://www.youtube.com/playlist?list=PLbHYdvrWBMxYGMvQ3QG6paNu7CuIRL5dX.
2. Barry Van Veen. All Signal Processing Channel on YouTube. https://www.youtube.com/user/allsignalprocessing.
3. Iman Moazzen. Signal Processing Hacks With Iman. http://www.sphackswithiman.com.
4. Iman Moazzen. YouTube Channel for Signal Processing Hacks With Iman. https://www.youtube.com/channel/UCVkatNMgkEdpWLhH0kBqqLw.
5. Wolfram Alpha Derivative Calculator. https://www.wolframalpha.com/input/?i=derivative+.
6. Wolfram Alpha Integral Calculator. https://www.wolframalpha.com/input/?i=integral+.
7. Wolfram Alpha Unilateral Laplace Transform Calculator. https://www.wolframalpha.com/input/?i=laplace+transform+calculator.

Online Resources II

8. Wolfram Alpha Unilateral Z Transform Calculator. https://www.wolframalpha.com/input/?i=Z+transform+calculator.
9. DSP Stack Exchange. https://dsp.stackexchange.com.
10. Math Stack Exchange. https://math.stackexchange.com.