BIEN350
Coursepack

Biosignals, Systems and Control

Prepared by BUSS

Professor Georgios Mitsis

Preface

Welcome to the BIEN350 — Biosignals, Systems and Control official course-pack! This document has been prepared by the Bioengineering Undergraduate Student Society (BUSS) to best assist you as you make your way through the course. The course-pack addresses the material according to an objective-based approach. This means that key concepts from each lecture are listed and expanded upon.

The course-pack is organized into four different parts:

- First, you will find tips and tricks for success in the course.
- Then, “primer” information has been included on complex numbers, MATLAB, and circuits. This will refresh your memory of topics that you should be familiar with as you begin the course.
- Finally, you will find course material information, such as explanations, notes, custom figures, and examples organized to directly respond to each lecture’s topics.

Please note that this document was created using source material from the Fall 2019 and Fall 2020 semesters. Due to the ever-changing nature of courses offered by the Department of Bioengineering at McGill, it is possible that parts of this course-pack no longer accurately reflect the current contents or organization of the course. In case of doubt, refer directly to the course instructor, Professor Georgios Mitsis.

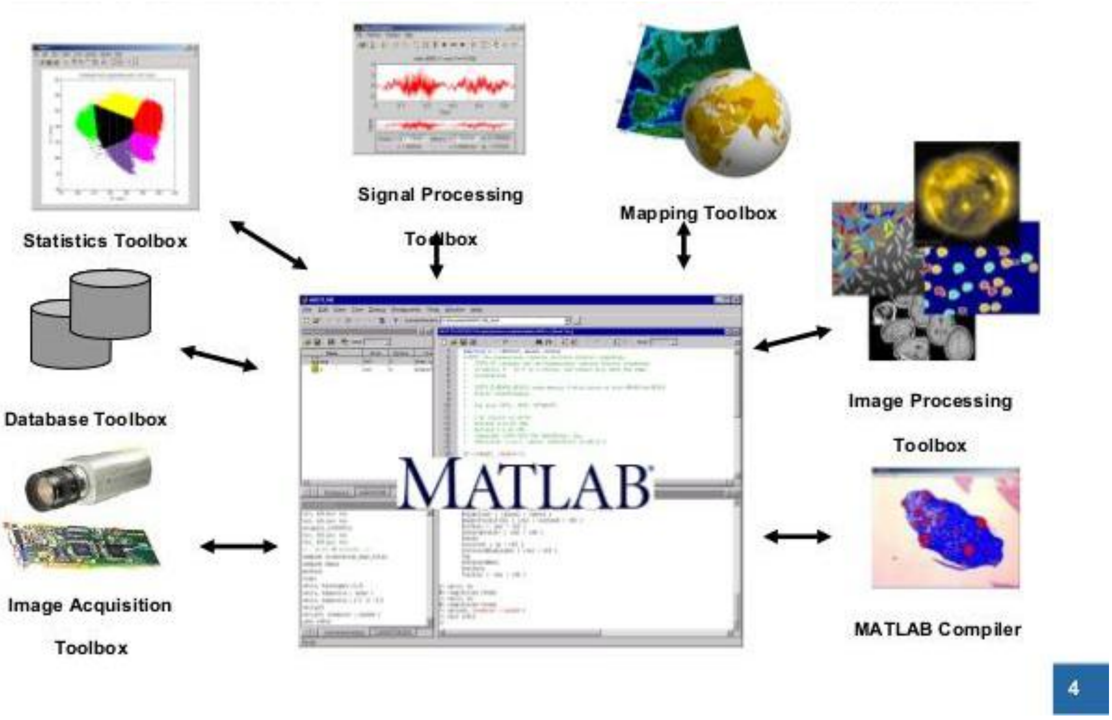
This course-pack could not have been produced without the help of outgoing BUSS Vice-President of Academics Anna Ciprick (2020 — 2021) and incoming BUSS Vice-President of Academics Jake Pringle (2021 — 2022).

Written and designed by Abdullatif Hassan and Alexander Becker
Reviewed by BUSS

MATLAB Primer

Introduction to MATLAB

MATLAB stands for **Mat**rix **Lab**oratory. It is a popular scientific research tool that is particularly useful for simulation applications. There exist various toolboxes for different research interests such as the statistics toolbox, the signal processing toolbox, the mapping toolbox, and the image processing toolbox.



Setup

Using MATLAB is free for McGill students. To download and install MATLAB, go to https://www.mathworks.com/login?uri=%2Fmwaccount%2F and start by creating a Mathworks account using your McGill email address. Once you have an account, follow the on-screen instructions to obtain your license and download MATLAB. As an aside, the following webpage lists all the software students and faculty have access to while at McGill: https://mcgill.service-now.com/itportal?id=kb_article&sysparm_article=KB0010741.

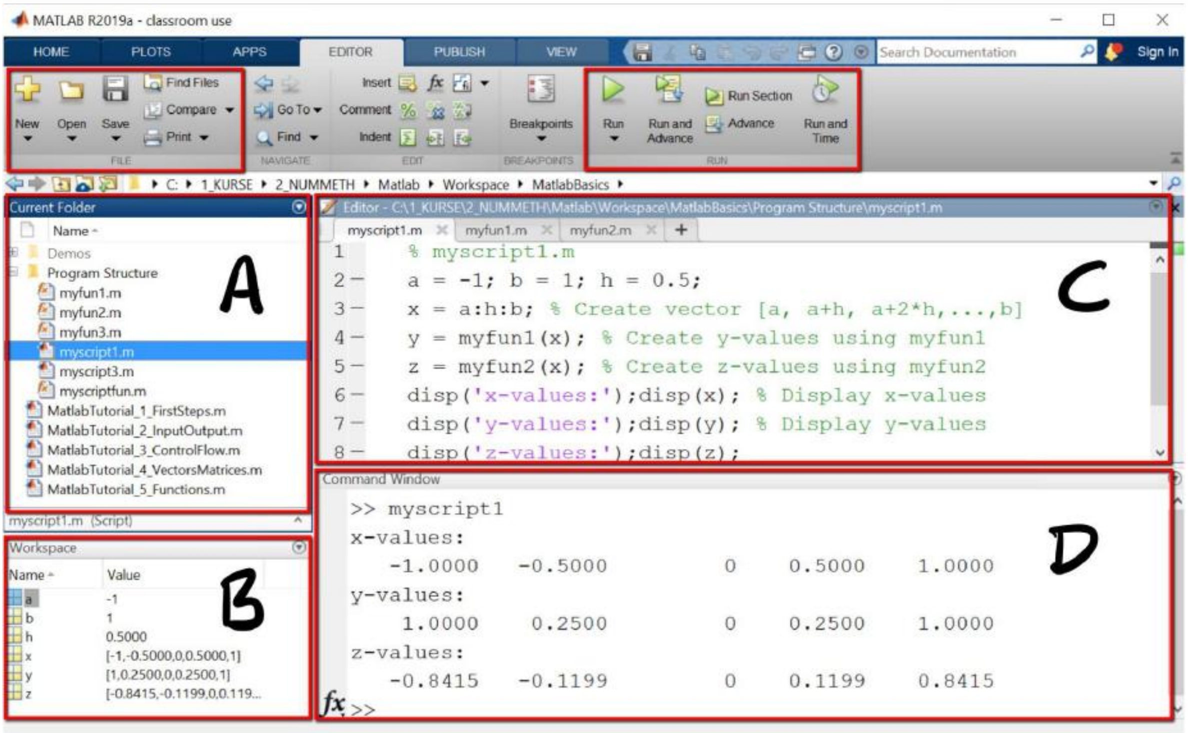
Once you have MATLAB installed on your computer, you are ready to start learning the basics. The following image shows the MATLAB integrated development environment (IDE) window. The configuration of the smaller "sub-windows" may be different for you, but they can be moved, resized, collapsed, and expanded, allowing you to create your preferred layout. The sub-windows are the following:

(A) The *current folder* pane shows the folder your script is currently located in. It can also allow you to browse the contents of your folder tree without leaving the MATLAB environment.

(B) The *workspace* pane lists all the variables currently loaded into RAM along with their values. Unlike many other popular programming languages, MATLAB's variables persist after the end of a program's execution. Variables must be intentionally cleared from the workspace, whether this be done manually or programmatically. Closing MATLAB will also clear the workspace.

(C) The *editor* pane allows you to create repeatedly-used scripts and functions. This is roughly equivalent to the environment you would use to write code in most other IDEs.

(D) The *command window* is used to type shorter commands and perform quicker operations. It also outputs the results of any commands or scripts. In this respect, it is similar to a command line console.



<u>Using MATLAB</u>

All variables in MATLAB are matrices. A single number, for example, is simply a 1x1 matrix.

- Vector initialization examples:
  - Row vector → R = [1,2,3,4] *or* R = [1 2 3 4]

```
R =

     1     2     3     4
```

  - Column vector → C = [1;2;3;4]

```
C =

     1
     2
     3
     4
```

- Matrix initialization example:

```
A =

     1     2
     3     4
```

Since MATLAB is based on matrices, it handles matrix operations very well.

- Matrix addition and subtraction (the matrices must be of the same size)

```
>> [1 2;3 4]+[1 2 ;3 4]

ans =

     2     4
     6     8
```

- Matrix multiplication will work if the matrices obey the matrix multiplication conditions.

```
>> [1 2;3 4]*[2 2 ;2 2]

ans =

     6     6
    14    14
```

- Element-wise multiplication (the matrices must be of the same size) can be accomplished as follows. Note that multiplying a matrix by a constant does not require the period.

```
>> [1 2;3 4].*[2 2 ;2 2]

ans =

     2     4
     6     8
```

- Matrix inversion: N = **inv(A)**
- Matrix transpose: T = A'
- Get the size of a matrix (including one-dimensional vectors): S = **size(A)**

To access information in MATLAB matrices, you can use parentheses along with the "location" of the data you would like to access. Contrary to most other programming languages, these locations (indices) in MATLAB begin at 1, not 0.

- Access one specific value: A(3,7) accesses the value at the third row, seventh column of matrix A
- Accessing multiple values can be done using the colon operator.
  - Access an entire row: A(3,:) accesses the entire third row (read as "row three, all columns").
  - Access an entire column: A(:,7) accesses the entire seventh column ("read as "column seven, all rows").
  - Access a range of data: A(2:5,7) accesses rows 2 through 5 (inclusively) of column 7.

The colon operator can also be used to define matrices.

- A = 1:10 → A = [1,2,3,4,5,6,7,8,9,10]
- A = 1:2:10 → A = [1,3,5,9], the 2 is used to "jump" to every second number in the range. Not including this "middle" number defaults its value to 1, as can be seen in the first example.

The **logspace(a,b,n)** and **linspace(a,b,n)** functions can be used to generate vectors of n elements logarithmically and linearly spaced, respectively, from a to b (inclusively).

Certain key MATLAB functions allow you to create special, very useful matrices.

- **zeros(m,n)** → create an mxn matrix of zeros (this is often used for memory preallocation when particularly large matrices are required)
- **ones(m,n)** → create an mxn matrix of ones (this is often used for memory preallocation when particularly large matrices are required)
- **eye(n)** → creates an nxn identity matrix

To make your code more understandable to readers, you can add comments. Comments start with a percentage symbol (%). Comments can span entire lines or be included at the end of a line of code. To comment out a line (turn a line of code into a comment), you can use the "ctrl+r" shortcut (or possibly the "ctrl+/" shortcut). To undo this action, you can use the "ctrl-t" shortcut.

When a choice is required in a script, if/else statements can be used. As a general rule, the format of such a statement is the following. Note that the "else if" and "else" branches are optional. Furthermore, multiple "else if" branches can be used.

```
if BOOLEAN_EXPRESSION
    CODE
else if OTHER_BOOLEAN_EXPRESSION
    CODE
else
    CODE
end
```

The boolean operators in MATLAB are similar to those used in other programming languages with a few exceptions.

- & → AND
- | → OR
- ~ → NOT
- ~= → NOT EQUALS
- == → EQUALS
- Logical 1 → TRUE
- Logical 0 → FALSE

Again, as is the case in most other programming languages, “for” and “while” loops are very popular.

- “For” loops (where **i** is incremented at every loop according to the range **1:n**)
  for i = 1:n
          CODE
  end
- “While” loops (the loop will continue as long as the **CONDITION** is true)
  while CONDITION
          CODE
  end

Besides being used for separating columns in matrix definitions, semicolons are used at the end of lines of code to suppress output. As MATLAB is an *interpreted* language, it executes each line of code as an independent command as the line is read. As such, the result of every line is output to the command window unless a semicolon is used to suppress that output. At a minimum, as good practice, typically:

- Use semicolons at the end of lines inside loops;
- Use semicolons at the end of lines that would otherwise output particularly large variables;
- Do not use a semicolon when the line computes the final result of interest to you.

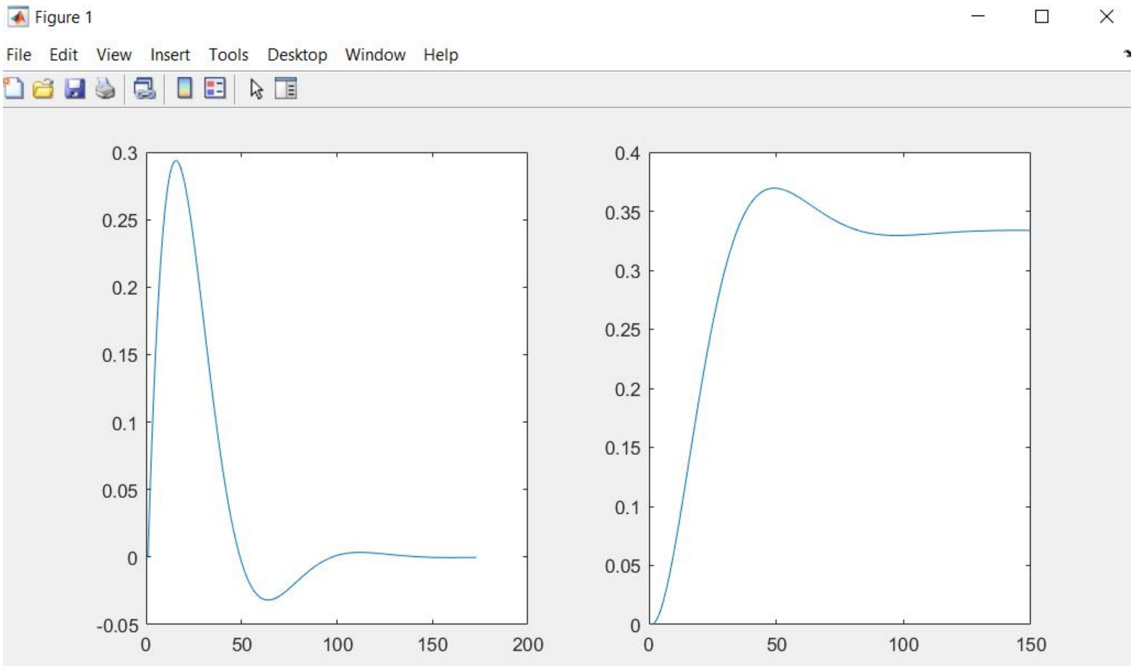
Plotting is a relatively simple task in MATLAB. The following are a few key commands to keep in mind.

- figure(n) → creates a new figure and assigns it number n (note that leaving the parentheses empty will simply increment the previous figure number by one)
- plot(x,y) → plots a connected graph of vector x versus vector y (note that these vectors must be of equal length)
- xlim([a b]) → sets the x axis bounds to start at a and end at b
- ylim([a b]) → sets the y axis bounds to start at a and end at b
- title(“title”) → sets the title of the graph to “title”
- xlabel(“x axis”) → sets the x-axis label to “x axis”
- ylabel(“y axis”) → sets the y-axis label to “y axis”

“Subplotting” can be used to group graphs together. The function subplot(m,n,p) divides the figure into an mxn grid of plots. “p” specifies the location on the grid of the plot function that follows it. For example:

- **figure(1)**
- **subplot(1,2,1)**
- **plot(x1,y1)**
- **subplot(1,2,2)**

- **plot(x2,y2)**



Other useful plotting functions include the following:

- **scatter(x,y)** → plots a scatter graph of vector x versus vector y
- **legend("A", "B", ...)** → assigns legend labels to the curves and distributions in the respective order they were added to the figure
- **text(x,y,"text")** → adds text to a figure at position (x,y)
- **hold on** → allows you to stack multiple curves and distributions onto a single figure

Functions are blocks of code that perform one particular task and that are often reused. MATLAB functions can be included at the end of scripts, or within their own .m files. Note that function files must be included in the same folder as the script that uses them. Furthermore, the name of the file must be the same as that of the function. The format for defining a function is the following:

function [output_variable1, …] = function_name(input_variable1, …)
        CODE
end

```
function [m,s] = stat(x)
    n = length(x);
    m = sum(x)/n;
    s = sqrt(sum((x-m).^2/n));
end
```

<u>MATLAB in BIEN350</u>

Plotting 2D time signals can be done as follows:

```
Fs = 8000;                    % samples per second
dt = 1/Fs;                    % seconds per sample
StopTime = 12;                % seconds
t = (0:dt:StopTime-dt);       % seconds
y = some function of t
plot(t, y)
```

The following steps can be used to create a linear regression:

(1) Use **scatter(x,y)** to plot the data
(2) Use **coefs = polyfit(x,y,1)** to obtain the slope and y-intercept of the line of best-fit
(3) Construct the line of best-fit by using **line = coefs(1)*x + coefs(2)**
(4) Plot the line of best-fit
(5) Use **fit = fitlm(x,y)** to obtain the linear best-fit model
(6) Use **fit.Rsquared.Ordinary** to obtain the R-squared value of the fit

The tf() function is a powerful tool for building systems like those used in BIEN350.
sys = tf([transfer function numerator],[transfer function denominator])

```
>> sys = tf([1], [1, 2, 3])


sys =


        1
  -------------
  s^2 + 2 s + 3
```

<u>Other useful MATLAB commands for BIEN350</u>

The following are a number of other commands (functions) that may be useful throughout BIEN350:

- **impulse(sys)** → input a transfer function and it outputs the impulse response of the system with that transfer function (the system **sys** is the output of the tf() function)
- **step(sys)** → same as impulse but outputs the step response
- **stem(X,Y)** → makes a discrete plot for Y as a function of X (stem(Y) also works, plotting Y on the unit x-axis)
- **freqs(numerator, denominator, w)** → establishes a frequency domain response for a transfer function with a certain numerator and denominator; w is the range of angles on the x-axis
- **abs(h)** → returns the magnitude plot of the frequency response; h must be the output of a freqs() function used previously
- **mag(h)** → returns magnitude of the frequency response **h**
- **freqz(numerator, denominator, n)** → similar to freqs() but in discrete time
- **bode(sys)** → plots the magnitude and phase bode diagrams of a system

- **[numerator, denominator] = butter(n, wc)** → returns the transfer function coefficients for a nth degree butterworth filter with cutoff frequency **wc**
- **[n,Wn] = buttord(Wp,Ws,Rp,Rs)** → returns the order and cutoff frequency of a butter filter that has the given input characteristics (outputs the inputs you need to subsequently pass to the butter function)
- **[numerator, denominator] = cheby1(n, Rp, wp)** → same as butter() but for the chebychev filter; note the different inputs; note that we can also use [z, p, k] as an output format instead of [numerator, denominator], giving the zero-pole-gain profile
- **[n, Wp] = cheb1ord(Wp, Ws, Rp, Rs)** → similar to buttord() but for the chebychev filter

Note that you can also consult the MATLAB documentation online for more information on the above functions, as well as any other function you may want to use. To access the documentation, go to https://www.mathworks.com/help/matlab/. If the official documentation doesn't answer all your questions, Google is your friend!

Complex Numbers Primer

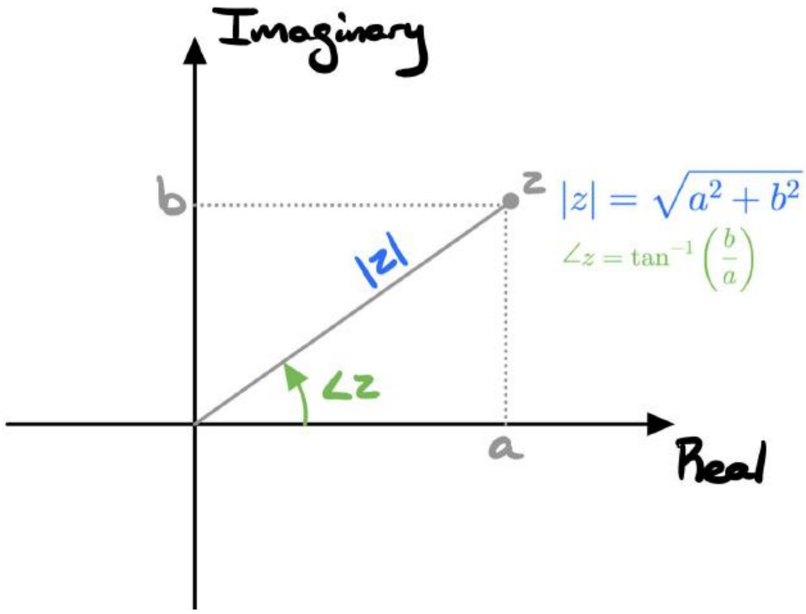
Introduction to complex numbers

Complex numbers are based on the imaginary number, often referred to as *i*, or *j* in engineering settings (to avoid confusion with the *î* unit vector).

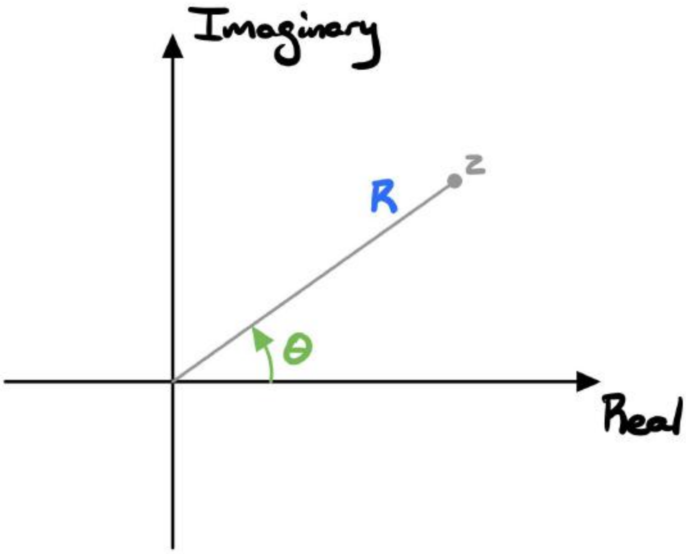
$j = \sqrt{-1}$

Complex numbers are denoted by *Z*. Contrary to real numbers, complex numbers have two components: magnitude and phase. These can be represented according to two different notations.

1) **Cartesian Notation**: $Z = a + jb$ where *a* and *b* are real numbers.
   - *a* is the real term
   - *jb* is the imaginary term
   - The magnitude of Z is given by $|Z| = \sqrt{a^2 + b^2}$
   - The phase of Z is given by $\angle Z = \tan^{-1}\left(\frac{b}{a}\right)$
   - Consider the two special cases:
     - Z is a purely real number when $b = 0$
     - Z is a purely imaginary number when $a = 0 \; and \, b \neq 0$



2) **Polar Notation**: $Z = Re^{j\theta}$ where *R* and *theta* are real numbers.
   - The magnitude of Z is given by $|Z| = R$
   - The phase of Z is given by $\angle Z = \theta$
   - Consider the two special cases:
     - Z is a purely real number when $\theta = 0 \; or \, \theta = \pi$
     - Z is a purely imaginary number when $\theta = \pm \pi/2$



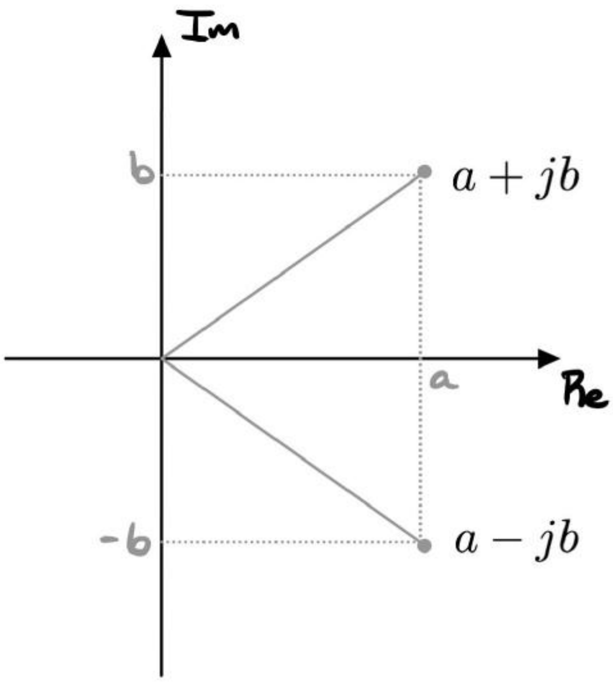
Properties of complex numbers

$Let\ Z_1,\ Z_2,\ and\ Z_3\ be\ complex\ numbers:$

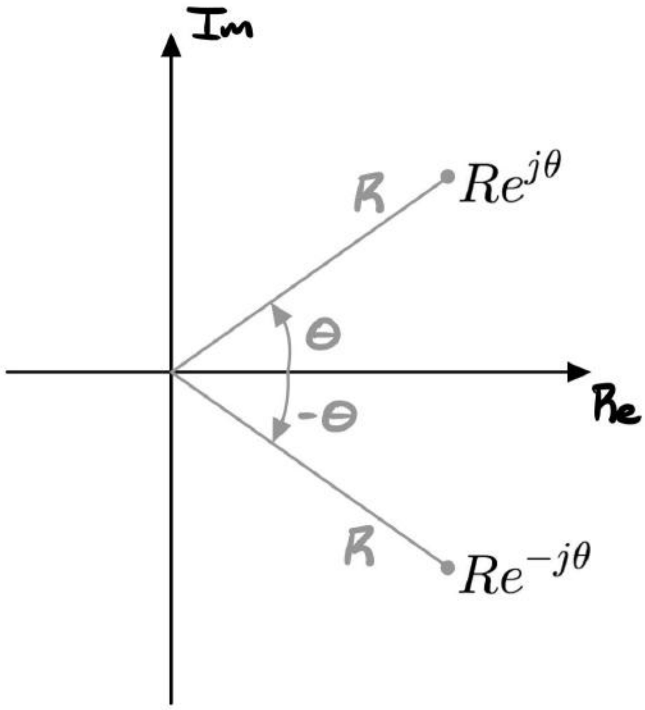
$$1)\ If\ Z_3 = Z_1 \times Z_2\ then\quad |Z_3| = |Z_1| \times |Z_2|$$
$$\angle Z_3 = \angle Z_1 + \angle Z_2$$
$$2)\ If\ Z_3 = \frac{Z_1}{Z_2}\ then\quad |Z_3| = \frac{|Z_1|}{|Z_2|}$$
$$\angle Z_3 = \angle Z_1 - \angle Z_2$$

Complex conjugates

1) In Cartesian notation, the complex conjugates have the same real part but opposite imaginary parts.



2) In polar notation, the complex numbers have the same magnitude but opposite phases.



Euler's formula

Euler's formula, also known as Euler's relation, allows us to transition between Cartesian and polar notations. It is defined as the following:

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

Consequently, we can derive two key equivalences for trigonometric functions.

$$\cos(\theta) = \frac{e^{j\theta} + e^{-j\theta}}{2}$$

$$sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j}$$

Introduction

Signal: A quantity (dependent variable) that changes with respect to another quantity(ies) (independent variable)

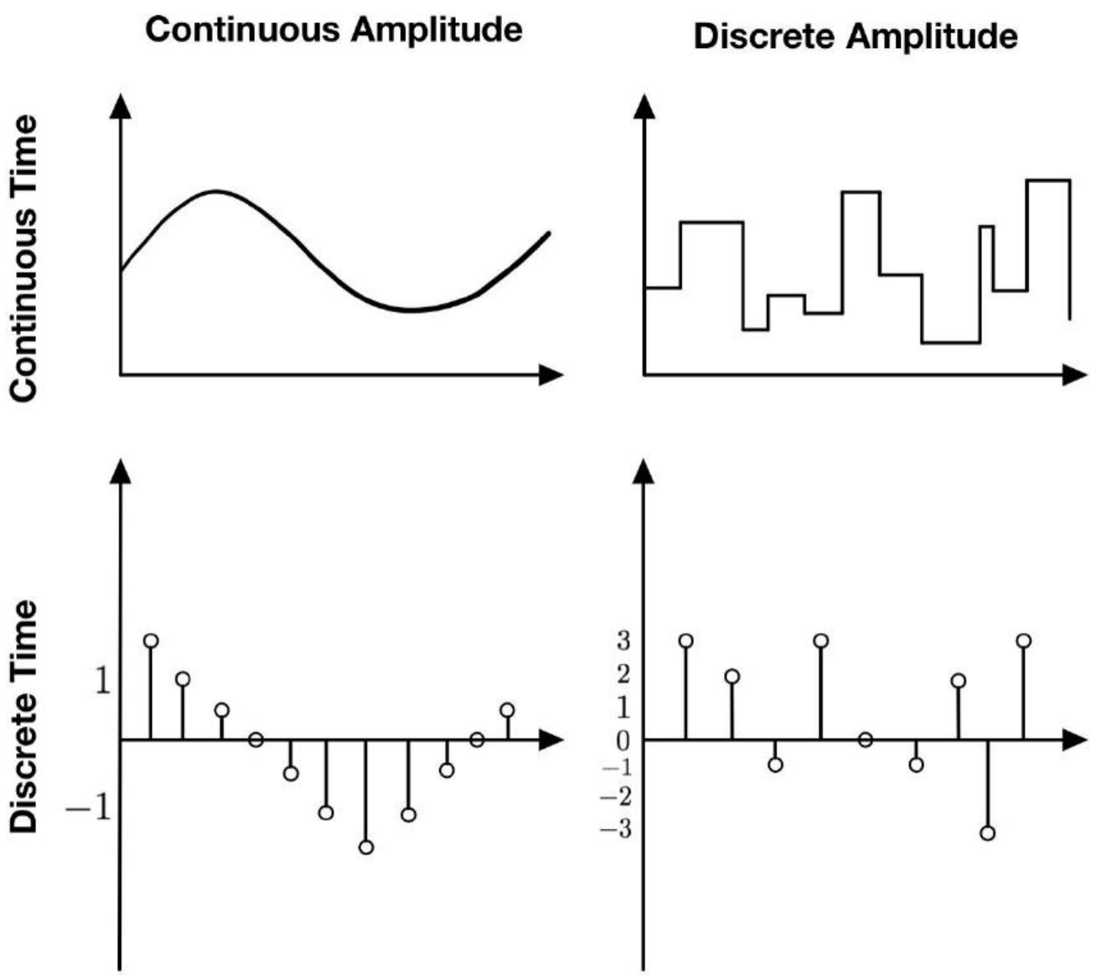
System: Any function that maps a set of inputs (x) to a set of outputs (y).

<u>Some types of signals:</u>

1) Temporal: Time is the independent variable (almost all signals you'll see in BIEN 350). Example: ECG
2) Spatial: Space is the independent variable. Example: Photograph
3) Spatiotemporal: Both time and space are independent. Example: Video

Digital vs Analog

Digital refers to a quantity being **discrete** whereas analog refers to a quantity being **continuous**.



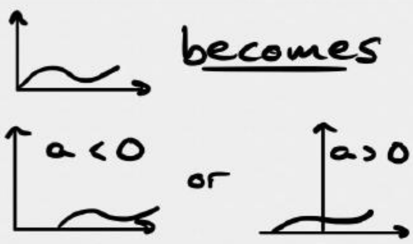
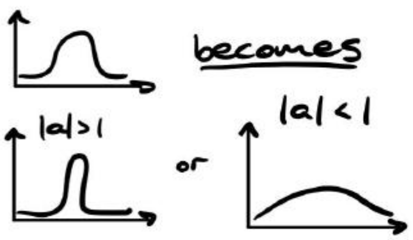
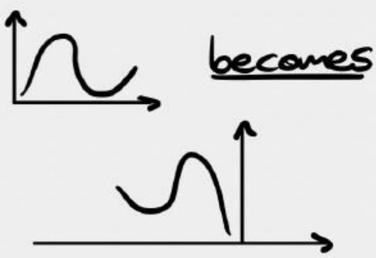
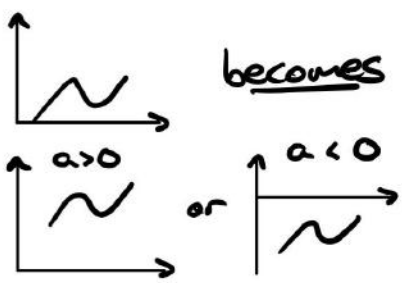
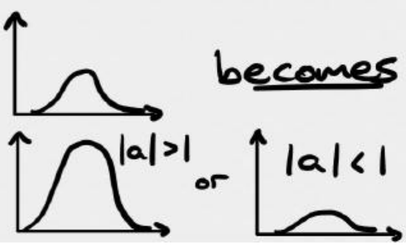
Time versus Frequency Domains

In BIEN 350, we deal with temporal signals in 2 different domains:

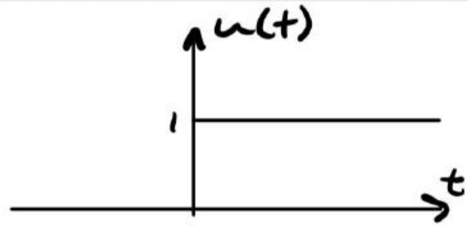
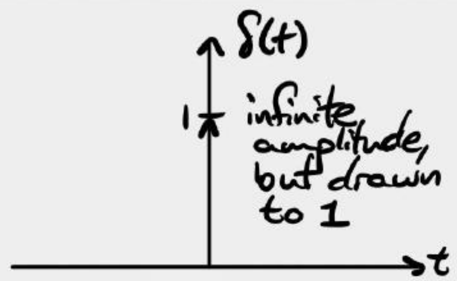
1) Time domain: Tells us how the signal behaves with respect to time (the original signal itself)
2) Frequency domain: Shows us the decomposition of this signal into different frequency waves and the strength of each of those frequencies.

We can convert between time and frequency domains using Laplace, Fourier, and Z-transforms (will appear later in the course)

Basic operations on signals

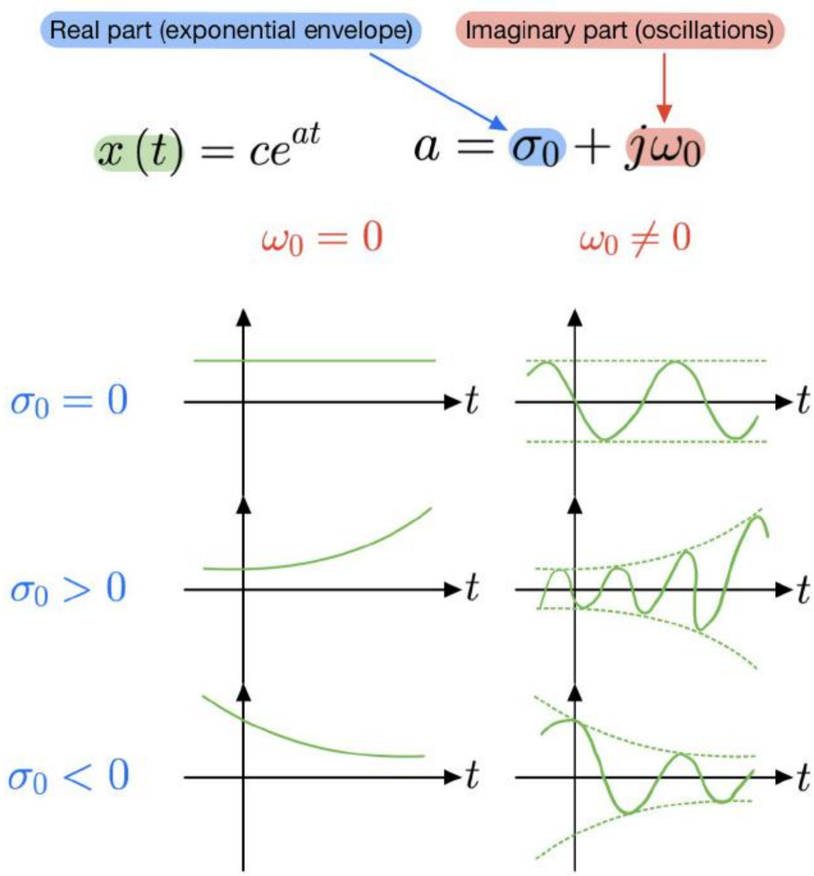
STEP WHEN SKETCHING	TYPE	FORM	CONDITIONS	GRAPHICALLY
#1	Shifting	y(t) = x(t + a), a real	a < 0 rightward a > 0 leftward	
#2	Scaling	y(t) = x(at), a real	|a| > 1 horizontal compression |a| < 1 horizontal expansion	
#3 (in any order after #1 and #2)	Flipping	y(t) = x(-t)	N/A	
#3 (in any order after #1 and #2)	Output Shifting	y(t) = x(t) + a, a real	a > 0 upward a < 0 downward	
#3 (in any order after #1 and #2)	Output Scaling	y(t) = ax(t), a real	|a| > 1 vertical expansion |a| < 1 vertical compression	

<u>Basic Categories of signals</u>

TYPE	FORM	NOTES	EXAMPLE or SKETCH
Even	$x(-t) = x(t)$	A function can be broken down into its even and odd components.	Given $x(t) = e^t$ Then $x_e(t) = Ev\{x(t)\} = 1/2(e^t + e^{-t})$, as $x_e(t) = Ev\{x(t)\} = 1/2[x(t) + x(-t)]$
Odd	$x(-t) = -x(t)$	A function can be broken down into its even and odd components.	Given $x(t) = e^t$ Then $x_o(t) = Od\{x(t)\} = 1/2(e^t - e^{-t})$, as $x_o(t) = Od\{x(t)\} = 1/2[x(t) - x(-t)]$
Periodic	$x(t) = x(t + T)$, period T (real for continuous, integer for discrete)	T_0 is the fundamental period, that is the smallest possible period value.	$x(t) = \sin(t)$ $x(t) = \cos(t)$
Step	$u(t) = 1 \ (t \geq 0)$ $u(t) = 0 \ (t < 0)$	Its derivative is the impulse function.	
Impulse	$\delta(t) = 0 \ (t \neq 0)$ $\int_{-\infty}^{\infty} \delta(t)dt = 1$	Infinite amplitude with unit energy. Its integral is the step function.	

CT Exponential Signals

A CT exponential time signal can be written in the following generic form:
$x(t) = ce^{at}$ $where\ a = \sigma_0 + j\omega_0$
The real part of a, is responsible for the exponential increase or decay of the signal whereas the imaginary part accounts for the oscillation which happens at frequency ω_0.



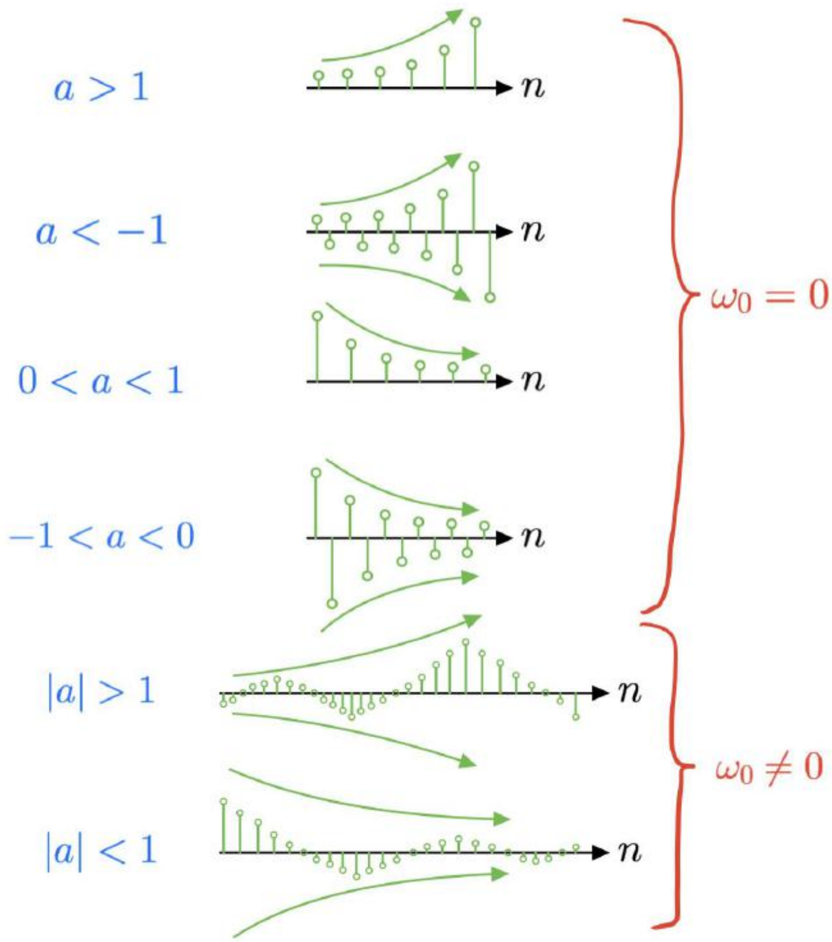
The fundamental period of an oscillating CT exponential signal is $T_0 = \frac{2\pi}{\omega_0}$.

DT Exponential Signals

A DT exponential signal can be written in the form $x[n] = Ca^n$ $where\ a = |a|e^{jw_0}$.

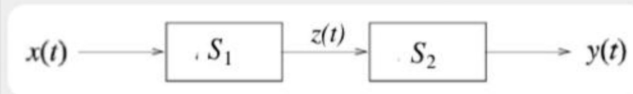
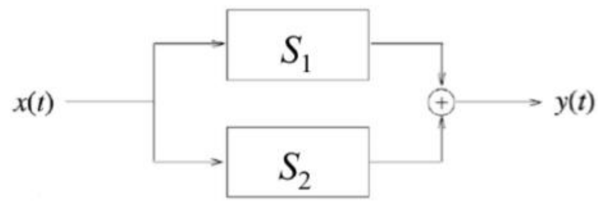
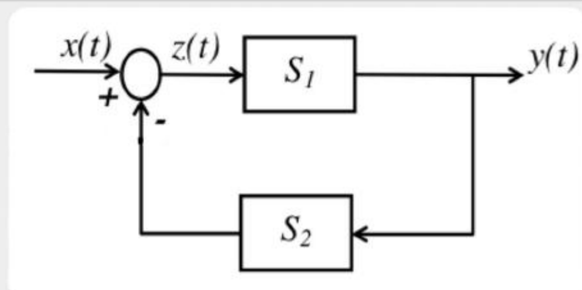
Similarly to CT exponential signals, the magnitude of ‘a’ characterizes the exponential behavior of the signal and the phase (imaginary part) characterizes the oscillations. Additionally, if ‘a’ is replaced by a negative real number, this would result in alternation behavior of the signal.

$$x[n] = ca^n \qquad a = |a|\, e^{j\omega_0}$$



The fundamental period of an oscillating DT exponential signal is $N = \frac{2\pi k}{\omega_0}$, where k is the smallest integer for which N is a whole number.

System Interconnections

TYPE	EQUATION	PRETTY PICTURE
Cascade (Series)	$y(t) = S_2[z(t)] = S_2[S_1[x(t)]]$	
Parallel	$y(t) = S_1[x(t)] + S_2[x(t)]$	
Feedback	Given $y(t) = S_1[z(t)]$ and $z(t) = x(t) - S_2[y(t)]$, then $y(t) = S_1[x(t) - S_2[y(t)]]$	

System Properties

I. Memory:
   A. Description:
      STATIC: Output only depends on input at present time (memoryless)
      DYNAMIC: Output depends on input at past and/or future inputs
   B. Math examples:
      STATIC: $y(t) = \sin(x(t))$
      DYNAMIC: $y(t) = \sin(x(2t))$ $\quad y(t) = \int_{t-1}^{t+2} x(t)dt$
   C. How to prove:
      1. Classic proof: to prove static, the input must be in the form of x(t) (i.e. the 't' inside should be intact) AND there should not be integrations over time. Otherwise the system is dynamic. Dynamic systems can also be proven by providing a counter example.
      2. Proof using the impulse response: $h(t) = 0 \; for \; all \; t \neq 0$ → static

II. Causality:

A. Description:

CAUSAL: Only depends on past or present inputs

NON-CAUSAL: Depends on future inputs

B. Math examples:

CAUSAL: $y(t) = x(t-1)$

NON-CAUSAL: $y(t) = x(t+1)$

C. How to prove:

1. Classic proof: To prove a causal system, the time values inside the parentheses of the input x() must be less than or equal to t AND in case there is an integral in the system, this integral must not cover any interval beyond t. Otherwise, the system is non-causal. Non-causal systems can be also proven by providing a counter example.
2. Proof using the impulse response: $h(t) = 0 \quad for\ all\ t > 0$ → causal

III. Stability (BIBO):

A. Description:

STABLE: If the input is bounded (i.e. finite), then the output must also be bounded (i.e. if $|x(t)| \leq A$, then $y(t) \leq B$)

UNSTABLE: Does not obey the above.

B. Math examples:

STABLE: $y(t) = x(t+2) - 3x(2t-1)$

UNSTABLE: $y(t) = \int_{-\infty}^{t} x(t)dt$

C. How to prove:

1. Classical proof: easier to prove unstable through a counter example. Stable systems can be proven by demonstrating in the system equation that the output stays bounded for bounded inputs.
2. Proof using the impulse response: $\int_{-\infty}^{\infty} |h(t)|\, dt < \infty$

IV. Invertibility:

A. Description:

INVERTIBLE: If x(t) maps to y(t), then knowing y(t), we can always find a unique x(t). In other words, there is a one-to-one mapping between the input and the output.

NON-INVERTIBLE: Does not obey the above

B. Math examples:

INVERTIBLE: $y(t) = 2x(t)$ $y(t) = \int_{-\infty}^{t} x(t)\, dt$

NON_INVERTIBLE: $y(t) = x^2(t))$ $y(t) = \frac{d}{dt} x(t)$

C. How to prove: Easier to prove non-invertible through a counter example. To prove invertible, one can find a unique form for x(t) in terms of y(t).

V. Time Variance:

A. Description:

TIME-INVARIANT: A timeshift in the input leads to the SAME timeshift in the output.

TIME-VARIANT: Does not obey the above.

B. Math examples:

TIME-INVARIANT: $y(t) = \cos(x(t))$

TIME-VARIANT: $y(t) = tx(t)$

C. How to prove:

Let $x_1(t) \rightarrow y_1(t)$

$x_2(t) = x_1(t-a)$

$x_2(t) \rightarrow y_2(t)$

If $y_2(t) = y_1(t-a)$ then the system is time invariant, otherwise it is time variant.

VI. Linearity:

A. Description:

LINEAR: Obeys superposition principles (additivity and scaling).

NON-LINEAR: Does not obey superposition principles.

B. Math examples:

LINEAR: $y(t) = ax(t)$

NON-LINEAR: $y(t) = x^2(t)$

C. How to prove:

Let $x_1(t) \rightarrow y_1(t)$

$x_2(t) \rightarrow y_2(t)$

$x_3(t) = ax_1(t) + bx_2(t)$ where a and b are constants

$x_3(t) \rightarrow y_3(t)$

If $y_3(t) = ay_1(t) + by_2(t)$ then the system is linear, otherwise it is non-linear.

Impulse and Step Responses

The **impulse response** h(t) or h[n] is the output that a system generates when the input is an impulse signal. The impulse response plays a key role in defining the system. Knowing the impulse response of a system, one can find the output of any input signal through convolutions or Fourier or Laplace transforms, which are seen later on in the course.

Similarly, the **step response** of a system s(t) or s[n] is the output generated from a step input signal. This also provides us with valuable information about the system.

As we have learned previously, there is a derivative-integral relation between the impulse signal δ(t), and the step signal u(t). The same can be said about the impulse and step responses of a signal.

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

$s(t) \ = \ \int_{-\infty}^{t} s(\tau)\, d\tau$ *Note that **τ** here is a dummy variable and that the final answer is in terms of t only.

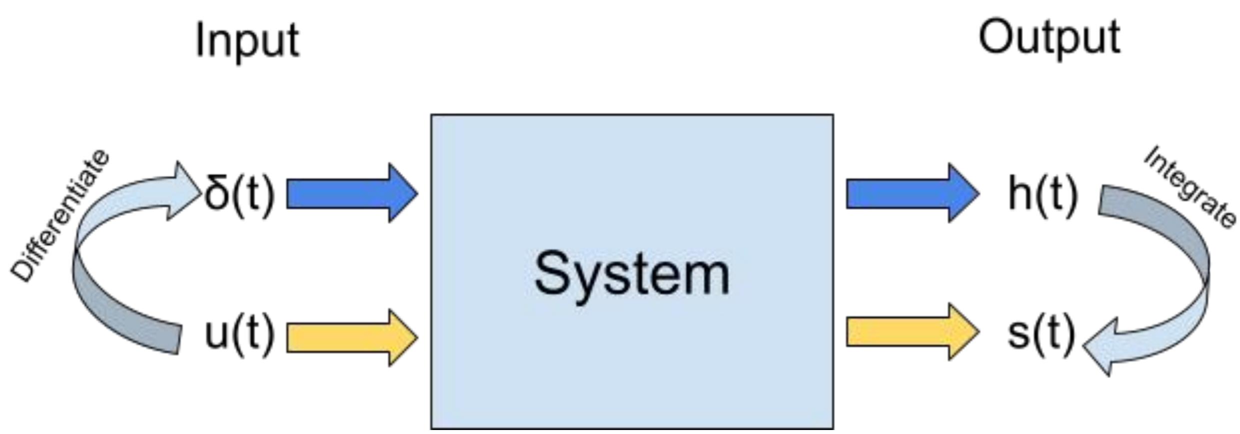
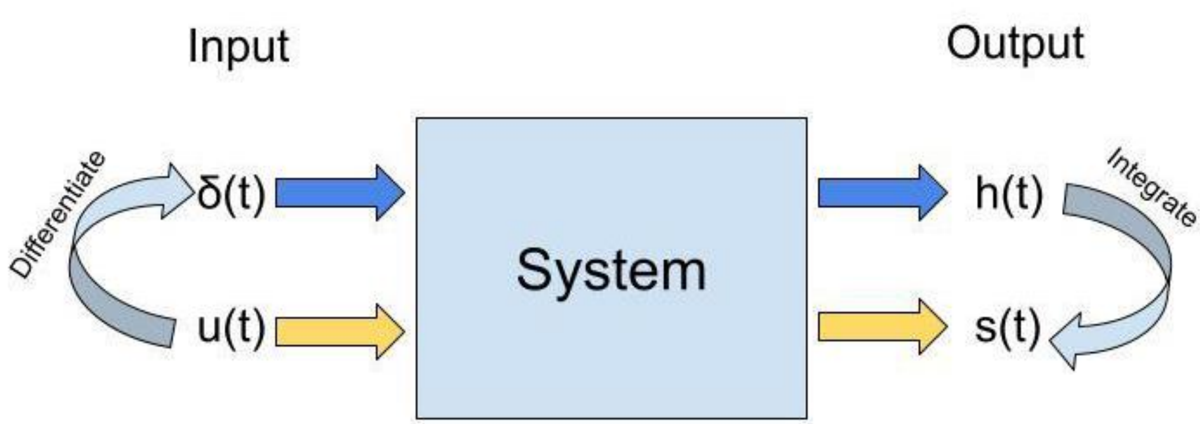


Image file because the original seems to be lacking all kinds of jpeg:



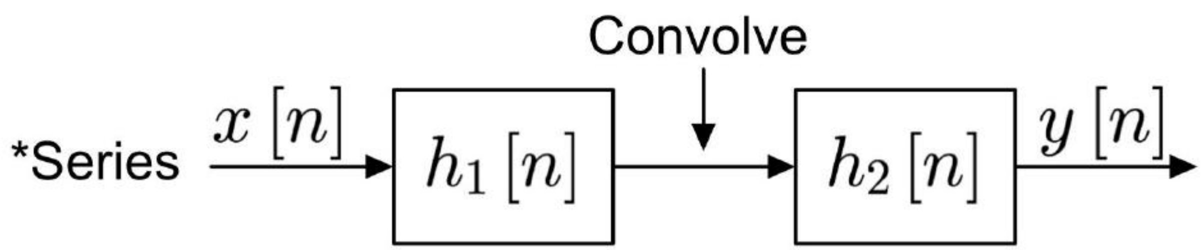
Convolutions Defined

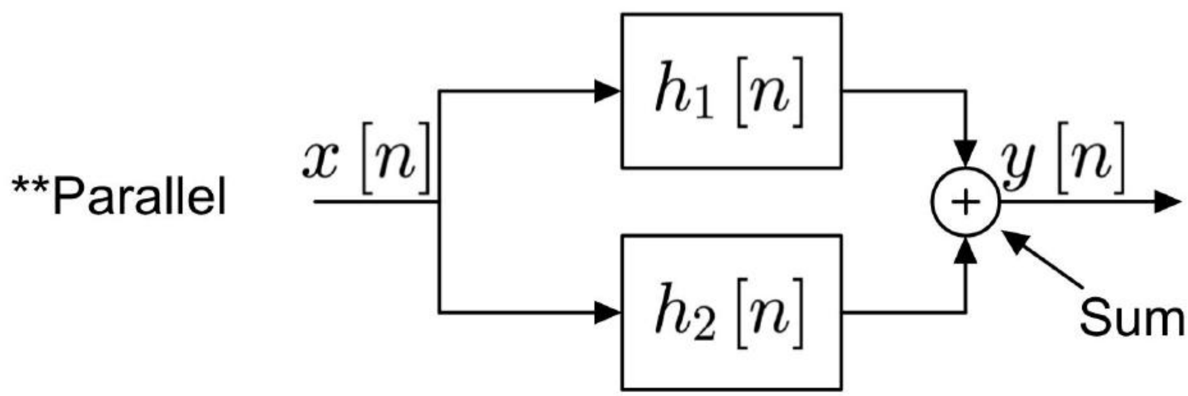
A convolution is a mathematical operation carried out between two signals. To understand convolutions fully, one must have both **graphical** and **mathematical** intuition of the concept.

Convolutions come in handy when they are used to determine the output of a system y(t) or y[n] to a given input x(t) or x[n]. This is done by performing a convolution between the input and the impulse response of the system (hence why the impulse response is an important property to know):

$$y(t) = x(t) * h(t)$$
$$y[n] = x[n] * h[n]$$

PROPERTY	CONTINUOUS FORM	DISCRETE FORM
Commutative	$x(t) \circledast h(t) = h(t) \circledast x(t)$	Exactly the same, but x(t) —> x[n] and h(t) —> h[n]
Associative	$x(t) \circledast (h_1(t) \circledast h_2(t)) = (x(t) \circledast h_1(t)) \circledast h_2(t)$	Exactly the same, but x(t) —> x[n] and h(t) —> h[n] (for all h)
Distributive	$x(t) \circledast (h_1(t) + h_2(t)) = x(t) \circledast h_1(t) + x(t) \circledast h_2(t)$	Exactly the same, but x(t) —> x[n] and h(t) —> h[n] (for all h)
Cascade (Series) Connection*	$h(t) = h_1(t) \circledast h_2(t)$	Exactly the same, but h(t) —> h[n] (for all h)
Parallel Connection**	$h(t) = h_1(t) + h_2(t)$	Exactly the same, but h(t) —> h[n] (for all h)
(with) Impulse Signals	$\delta(t - t_0) \circledast g(t) = g(t - t_0)$	$\delta[n - n_d] \circledast x[n] = x[n - n_d]$





Convolution Steps

Mathematically, the convolution of the input signal and an impulse response is performed by first shifting the impulse response, flipping it along the vertical axis, multiplying it by the input signal, and, finally, integrating over the entire domain.

$$h(\tau) \xrightarrow{\text{shift}} h(t+\tau) \xrightarrow{\text{flip}} h(t-\tau)$$
$$\xrightarrow{\text{multiply}} x(\tau)h(t-\tau) \xrightarrow{\text{integrate}} \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

The equation for the convolution of an input signal, x(t), and an impulse response, h(t), is, therefore, the following:

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

CT Convolution Example

Figure 1: We start off with two signals x(t) and h(t). Our aim here is to find the output of the system, y(t). To achieve this, we will perform a convolution between x(t) and h(t). Graphically, performing the convolution involves manipulating one of the signals while keeping the other one in place. It does not matter which signal is manipulated and which one is kept static due to the commutative property of convolutions. For the purpose of this example (and in accordance with the mathematical description given above), h(t) will be manipulated and x(t) will be the static signal.

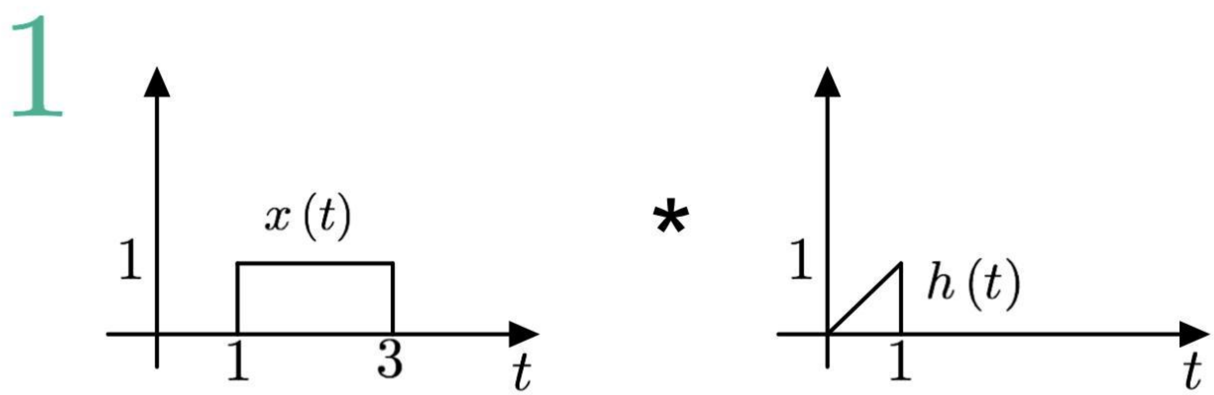


Figure 2 (**shift**): We start by replacing the t-axis by the τ-axis. The τ variable here is a temporary (dummy) variable. We use it during the convolution operation, but the final answer must not contain τ. The static signal is kept as x(τ) while the manipulated signal becomes h(t+τ). Once the t-axis is replaced, the t-variable translates into a shift along the τ axis. For example, if a point was on coordinate 0 of the t-axis, it comes out as a -t on the τ axis. A point that was 1 becomes 1-t (or -t+1, as shown in the figure) and so on. Note that it is possible to shift in the other direction (+t) but by convention, shifting by -t is used here.

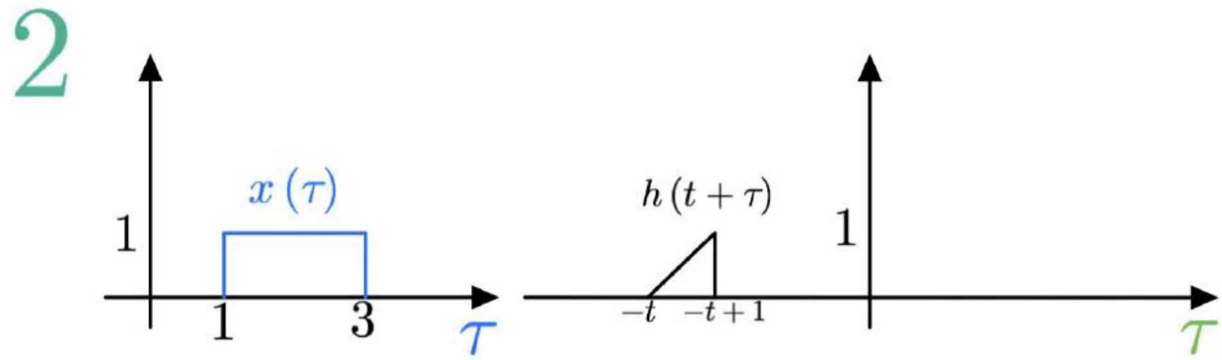


Figure 3 (**flip**): The manipulated signal h(t+τ) is transformed into h(t-τ) by reflecting it along the vertical axis. Graphically, the τ-coordinate of each point of the manipulated signal is multiplied by -1. For example, a coordinate of 1-t becomes -1+t (or t-1, as shown in the figure).

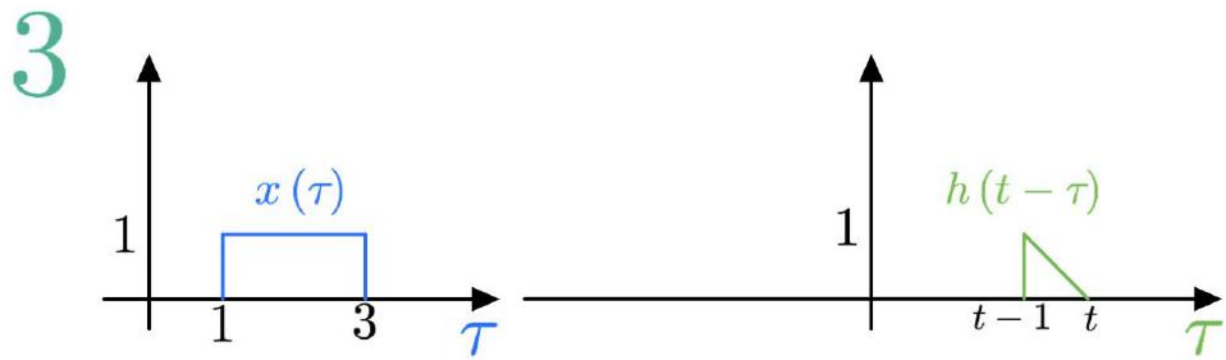


Figure 4 (**multiply** and **integrate** over the <u>first interval</u>): By choosing a value of t, the manipulated signal is anchored on to a fixed location on the τ-axis. We consider different ranges of t-values and, for each, we perform the multiply-and-integrate step of convolution. For the first interval, $t < 1$, we observe graphically that there is no overlap between the signals. Mathematically, this translates into a zero multiplication which results in a final answer of $y(t) = 0$ for $t < 1$.

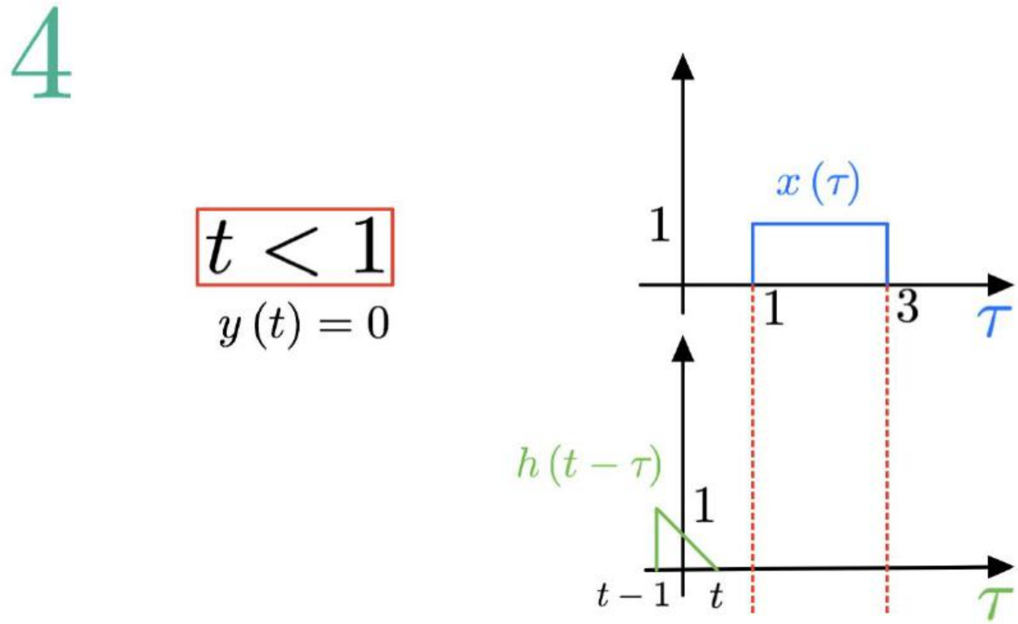


Figure 5 (**multiply** and **integrate** over the second interval): For 1 ≤ t ≤ 2, part of the manipulated signal, h(t-τ), falls in the non-zero interval of the static signal, x(τ). We thus perform multiplication and integration over the common interval. Graphically, this could be done by finding the area of the shape that results from multiplying the signals. Mathematically, the answer could be found by evaluating the integral with the bounds of integration set to the overlapping interval. In this case, the lower bound is fixed (1) and the upper bound is variable (t).

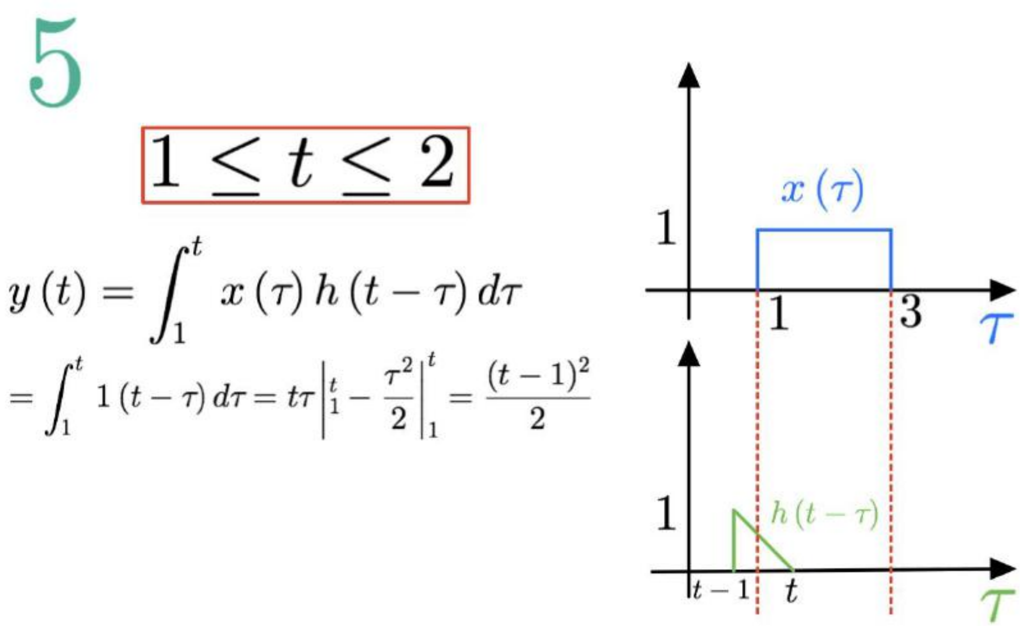


Figure 6 (**multiply** and **integrate** over the third interval): When $2 \le t \le 3$, the manipulated signal fully overlaps with the static signal. The convolution output can be calculated by either graphically finding the area of the shape resulting from multiplication, or by evaluating the integral with both bounds being in terms of variables (t and t-1)

6

$$2 \le t \le 3$$

$$y(t) = \int_{t-1}^{t} x(\tau)\, h(t-\tau)\, d\tau$$

$$= \int_{t-1}^{t} 1\,(t-\tau)\, d\tau = t\tau\Big|_{t-1}^{t} - \frac{\tau^2}{2}\Big|_{t-1}^{t} = \frac{1}{2}$$

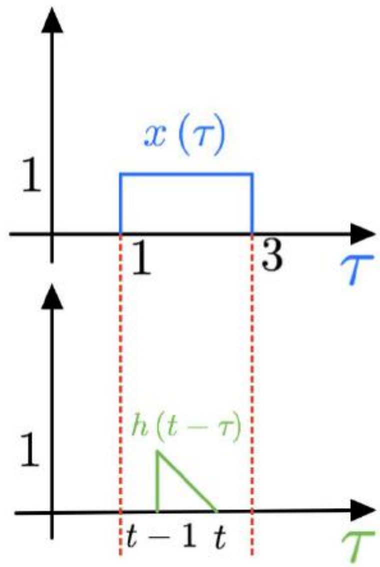


Figure 7 (**multiply** and **integrate** over the fourth interval): Similarly to what was seen in figure 5, there is partial overlap between the signals over the interval $3 \le t \le 4$. The difference here is that the upper bound is fixed while the lower bound is variable.

7

$$3 \le t \le 4$$

$$y(t) = \int_{t-1}^{3} x(\tau)\, h(t-\tau)\, d\tau$$

$$= \int_{t-1}^{3} 1\,(t-\tau)\, d\tau = t\tau\Big|_{t-1}^{3} - \frac{\tau^2}{2}\Big|_{t-1}^{3} = \frac{1}{2} - \frac{(t-3)^2}{2}$$

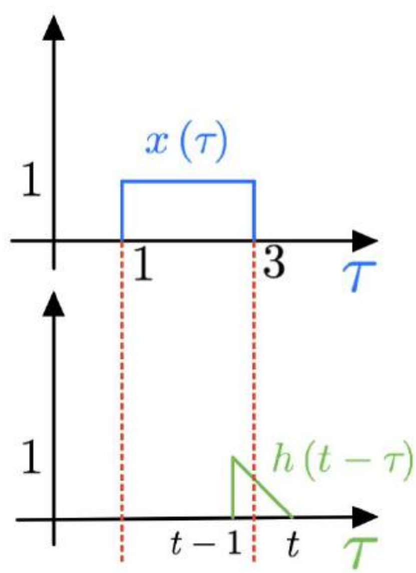


Figure 8 (**multiply** and **integrate** over the <u>fifth interval</u>): Just as in figure 4, there is no overlap between the signals over the interval t > 4. As such, the output evaluates to zero over this interval.

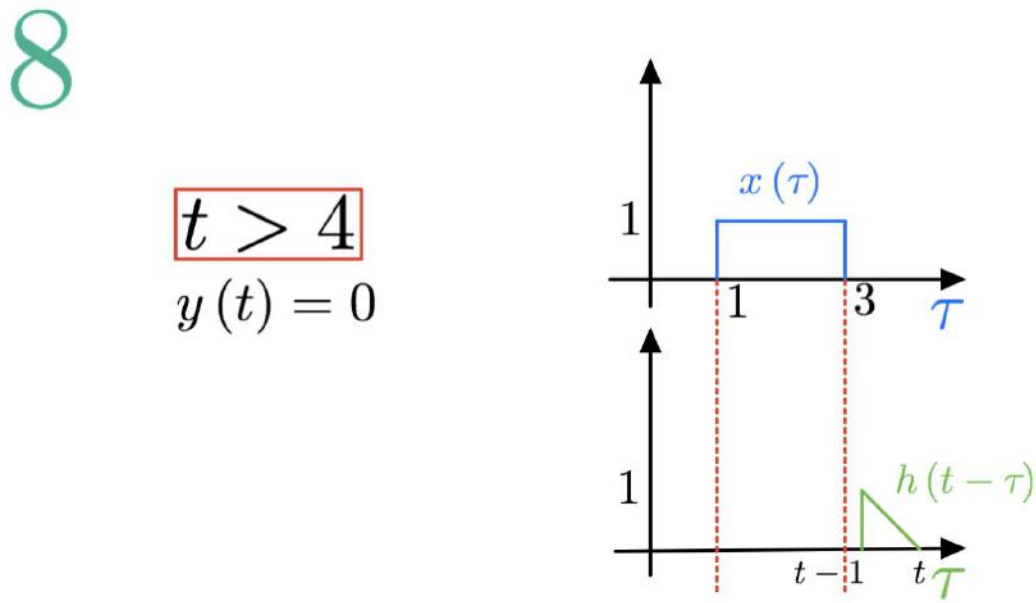
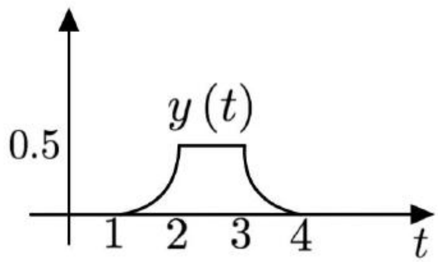


Figure 9 (resulting output signal): From the answers found in each of the intervals (figures 4 through 8), the final output, y(t), is constructed as a piecewise function.

9



$$y(t) = \begin{cases} 0 & t < 1 \\ \dfrac{(t-1)^2}{2} & 1 \le t \le 2 \\ \dfrac{1}{2} & 2 \le t \le 3 \\ \dfrac{1}{2} - \dfrac{(t-3)^2}{2} & 3 \le t \le 4 \\ 0 & t > 4 \end{cases}$$

Comment on DT Convolution

In the case of discrete signals, convolution is simpler to perform. Instead of an integral, the output signal, y[n], is given by the following equation:

$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$

Graphically, this results in the multiplication of the values of each “stem” of the discrete signals (the input signal and the impulse response) at each “frame” as the manipulated signal, h[n-k], slides past the static signal, x[n]. For each of these frames, the products of each multiplication are summed to form the stem of the output at each discrete time point, n.

Models and Systems

Modelling is a practice used to approximate a certain system. As a rule of thumb, the simplest model that can adequately represent the system is chosen. This is known as *Occam's Razor*. Ordinary Differential Equations (ODEs) are a common tool used to model certain LTI systems (such as those covered in this course).

Ordinary Differential Equations

General form (for BIEN 350): $ay''(t) + by'(t) + cy(t) = dx(t)$

We will use the following class example:
$y'(t) + y(t) = x(t)\,,\, x(t) = \cos(t)\,,\, y(0) = 2$

Solving using the y_p/y_h method from MATH263:

1. Find the roots of the polynomial $ar^2 + br + c$
   $r + 1 = 0 \rightarrow r = -1$
   Note: A system is stable only when all of the roots of the polynomial have a negative real part.

2. Using the roots found, determine the general form of the homogenous solution y_h
   $y_h = D\,e^{-t}\, where\, D = cst$

3. Using x(t), find the general form for the particular solution y_p
   $y_p = C_1 \sin(t) + C_2 \cos(t)$

4. Find the first and second order derivatives for y_p (generalized) then plug them into the original ODE to find the full form of y_p.
   $y_p' = C_1 \cos(t) - C_2 \sin(t)$
   $C_1 \cos(t) - C_2 \sin(t) + C_1 \sin(t) + C_2 \cos(t) = \cos(t)$
   $C_1 = C_2 = 0.5$
   $\rightarrow y_p = 0.5 \sin(t) + 0.5 \cos(t)$

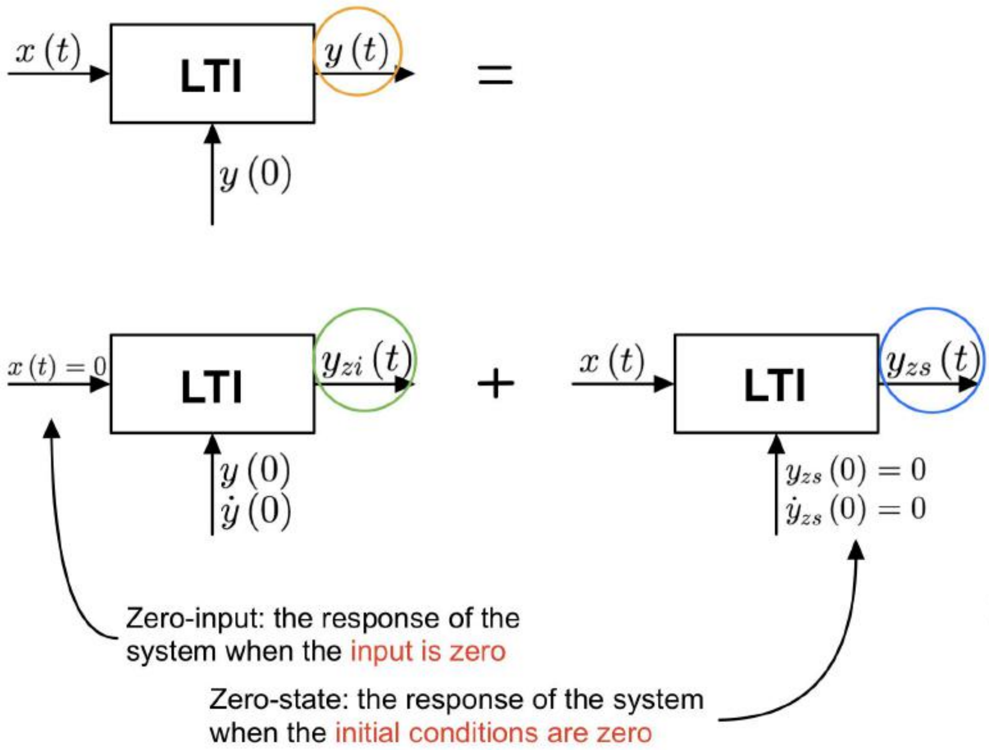
5. Merge the homogeneous and particular solutions ($y = y_h+y_p$)
   $y = De^{-t} + 0.5 \sin(t) + 0.5 \cos(t)$

6. Plug in the initial conditions to get rid of the coefficients in the y_h part of the solution
   $y(0) = 2$
   $2 = De^{-0} + 0.5 \sin(0) + 0.5 \cos(0)$
   $D = 1.5$
   $y = 1.5e^{-t} + 0.5 \sin(t) + 0.5 \cos(t)$

Zero-Input and Zero-State Method for BIEN350

This method is used for LTI systems only. Instead of breaking the solution into y_h and y_p, we use $y = y_{zi} + y_{zs}$. Zero-input (y_{zi}) is when **x(t)** = 0, but the **initial conditions** are non-zero. Zero-state is when the **initial conditions** are equated to zero, and **x(t)** is non-zero.

$$y(t) = y_{zi}(t) + y_{zs}(t)$$



To solve an ODE using this method:

1. Find the roots of the characteristic polynomial $ar^2 + br + c$
   $r + 1 = 0 \rightarrow r = -1$

2. Find the general form for y_{zi} which is the same as the y_h from general form
   $y_{zi} = De^{-t}$

3. Plug in the initial conditions to find the coefficients of y_{zi}
   $y_{zi}(0) = 2$
   $2 = De^{-0}$
   $D = 2$

4. Find the general form of y_p from the input
   $y_p = C_1 \sin(t) + C_2 \cos(t)$

5. Find the first- and second-order derivatives for y_p (generalized), then plug them into the original ODE to find the full form of y_p

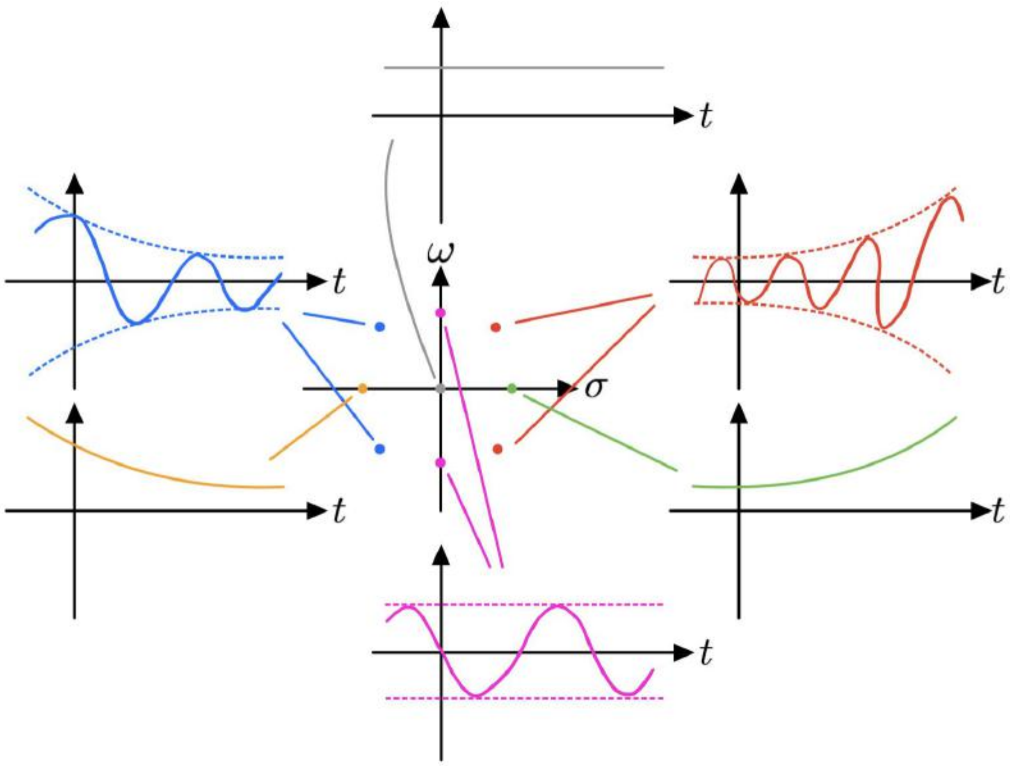
$$y_p' = C_1 \cos(t) - C_2 \sin(t)$$
$$C_1 \cos(t) - C_2 \sin(t) + C_1 \sin(t) + C_2 \cos(t) = \cos(t)$$
$$C_1 = C_2 = 0.5$$
$$\rightarrow y_p = 0.5 \sin(t) + 0.5 \cos(t)$$

6. Write the zero-state solution as a linear combination of the general form of the homogenous solution and the full particular solution $y_{zs} = y_h + y_p$
$$y_{zs} = D_2 e^{-t} + 0.5 \sin(t) + 0.5 \cos(t)$$

7. Using initial zero conditions, find the coefficients of the y_h component inside the zero-state solution
$$y_{zs}(0) = 0$$
$$0 = D_2 e^{-0} + 0.5 \sin(0) + 0.5 \cos(0)$$
$$D_2 = -0.5$$
$$y_{zs} = -0.5e^{-t} + 0.5 \sin(t) + 0.5 \cos(t)$$

8. Combine zero-input and zero-state solutions to form the final solution
$$y = y_{zi} + y_{zs}$$
$$y_{zs} = 2e^{-t} - 0.5e^{-t} + 0.5 \sin(t) + 0.5 \cos(t)$$
$$y = 1.5e^{-t} + 0.5 \sin(t) + 0.5 \cos(t)$$

Note: The main difference between these two approaches is when to plug in the initial conditions of the system. For the zero-input/zero-state method, plugging in happens to y_{zi} and y_{zs} individually, and both solutions are combined at the end. The method which uses y_h and y_p involves combining both solutions and then plugging in initial conditions.



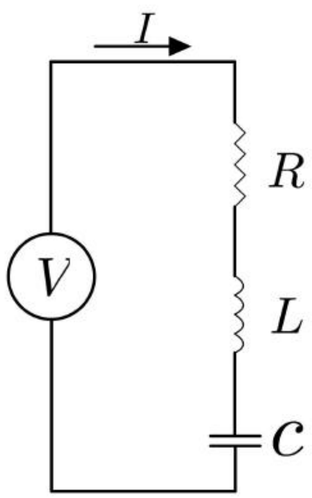
The zero-input response solution forms are summarized in the following table:

Root type	Homogeneous solution term
Real	$Ce^{\lambda_i t}$
Real with multiplicity m	$C_1 e^{\lambda_i t} + C_2 t e^{\lambda_i t} + \ldots + C_m t^{m-1} e^{\lambda_i t}$
Complex conjugate $\sigma_i \pm j\omega_i$	$e^{\sigma t}(C_1 \cos(\omega_i t) + C_2 \sin(\omega_i t))$
Imaginary conjugate $\pm j\omega_i$	$C_1 \cos(\omega_i t) + C_2 \sin(\omega_i t)$

The zero-state response solution forms are summarized in the following table:

Input form	Particular solution form
$x(t) = K$	$y_P(t) = C$
$x(t) = \beta e^{\alpha t}$	$y_P(t) = Ce^{\alpha t}$
$x(t) = \beta\cos(\omega t)$ $x(t) = \beta\sin(\omega t)$	$y_P(t) = C_1\cos(\omega t) + C_2\sin(\omega t)$
$x(t) = \beta_k t^k + \beta_{k-1}t^{k-1} + \cdots + \beta_0$	$y_P(t) = C_k t^k + C_{k-1}t^{k-1} + \cdots + C_0$ If β_0 is nonzero, otherwise $y_P(t) = t(C_k t^k + C_{k-1}t^{k-1} + \cdots + C_0)$
$x(t) = (\beta_k t^k + \beta_{k-1}t^{k-1} + \cdots + \beta_0)\, e^{\alpha t}$	$y_P(t) = (C_k t^k + C_{k-1}t^{k-1} + \cdots + C_0)e^{\alpha t}$ If β_0 is nonzero, otherwise $y_P(t) = t(C_k t^k + C_{k-1}t^{k-1} + \cdots + C_0)\, e^{\alpha t}$
$x(t) = (\beta_k t^k + \beta_{k-1}t^{k-1} + \cdots + \beta_0)\cos(\omega t)$	$y_P(t) = (C_k t^k + C_{k-1}t^{k-1} + \cdots + C_0)\cos(\omega t)$ If β_0 is nonzero, otherwise $y_P(t) = t(C_k t^k + C_{k-1}t^{k-1} + \cdots + C_0)\cos(\omega t)$

Solving a Circuit Problem



1. Identify the input and output of the system.

2. Apply one of the two fundamental circuit laws:
   a) Additivity of voltages (Kirchhoff's Voltage Law):
   $V_{power\ source} = \sum V_{components}$
   b) Junction law (Kirchhoff's Current Law):
   $\sum i_{in} = \sum i_{out}$

3. Write down the voltage or current equation for each electrical component:
   Resistor: $V_R(t) = Ri(t)$
   Inductor: $V_L(t) = L\frac{di(t)}{dt}$
   Capacitor: $V_c(t) = \frac{1}{C}\int_0^t i(\tau)d\tau$ or $i(t) = C\frac{dV_c(t)}{dt}$

4. Plug in the component laws into the equation generated by step two.

5. Simplify the equation so that the only variables it contains are the input, the output, and their derivatives.

6. Solve the resulting ODE.

Note: In physiological modelling, circuits are often used to represent physiological systems. It is common to see voltage representing pressure and to see current representing flow quantities (this is discussed further in BIEN462).

Introduction to Difference Equations

In nature, most systems exist as continuous time systems. On the other hand, computers and electronics are digitized systems that operate in discrete time. Thus, there is a need to discretize those CT systems. Discretization, the process of approximating the CT system as a DT system, can be done through one of the following ways:

A) Forward Euler Method: $\frac{dy}{dt} \approx \frac{y(t+h) - y(t)}{h}$ where h is the time-step used in the approximation.

B) Backward Euler Method: $\frac{dy}{dt} \approx \frac{y(t) - y(t-h)}{h}$

Solving Difference Equations

Overall, there exists some similarity in the general approach for solving CT ODE's and DT difference equations (DEs). When solving LTI DT systems, we assume that the input starts at n=0 and that when n<0, x[n] = 0. Thus, when given conditions for n<0 such as y[-1], y[-2], etc. one can find the zero-input response of the system. In this case, the given conditions are known as **initial conditions**. However, conditions can sometimes be given as y[0], y[1], y[2], etc. These conditions are known as **auxiliary conditions**. In this case, when applying the conditions, one cannot assume that x[n]=0. Hence, the answer found would not be a zero-input solution. As such, the approaches for solving a DE when given initial conditions and solving a DE when given auxiliary conditions differ.

<u>Steps to solve a DE:</u>

A) Given initial conditions: (Use zero-input/zero-state approach)

Example: $y[n] - 0.5y[n-1] = x[n]\,,\ x[n] = n^2u[n],\ y[-1] = 16$

1. Find the N roots of the characteristic polynomial:
   $r - 0.5 = 0,\ r = 0.5$
2. Using the roots, determine the general form of the homogeneous solution:
   $y_h[n] = \sum_{m=1}^{N} A_m Z_m^n \ where\ Z_m\ is\ each\ root$
   $y_h[n] = A\,0.5^n$
3. Apply the initial conditions to find the coefficients of the zero-input solution (in the case of multiple roots, we must solve a system of N equations with N unknowns to find the unknown coefficients):
   $16 = A \times 0.5^{-1} \rightarrow A = 8$
4. Find the particular solution:
   $y_p[n] = C2^n$
5. Plug the values into the DE to find the full particular solution:
   $C \times 2^n - 0.5C \times 2^{n-1} = 2^n \rightarrow C = 4/3$
6. Write the zero-state solution as a linear combination of y_p and y_h:
   $y_{zs}[n] = A\,0.5^n + \frac{4}{3}2^n$
7. Use the relations $y_{zs}[-1] = 0,\ y_{zs}[-2] = 0\cdots$ to find the zero-state solution at n=0, 1, …, N-1:

$y[0] - 0.5y[-1] = 1 \rightarrow y[0] = 9$

8. Using the zero-state solutions at n=0,1, …, N-1, find the unknown coefficients of the zero-state solution:
   $9 = A0.5^0 + \frac{4}{3}2^0 \rightarrow A = 23/3$
9. Combine the zero-input and zero-state solutions to obtain the full solution:
   $y[n] = 8 \times 0.5^n + \frac{23}{3}0.5^n + \frac{4}{3}2^n = \frac{47}{3}0.5^n + \frac{4}{3}2^n$

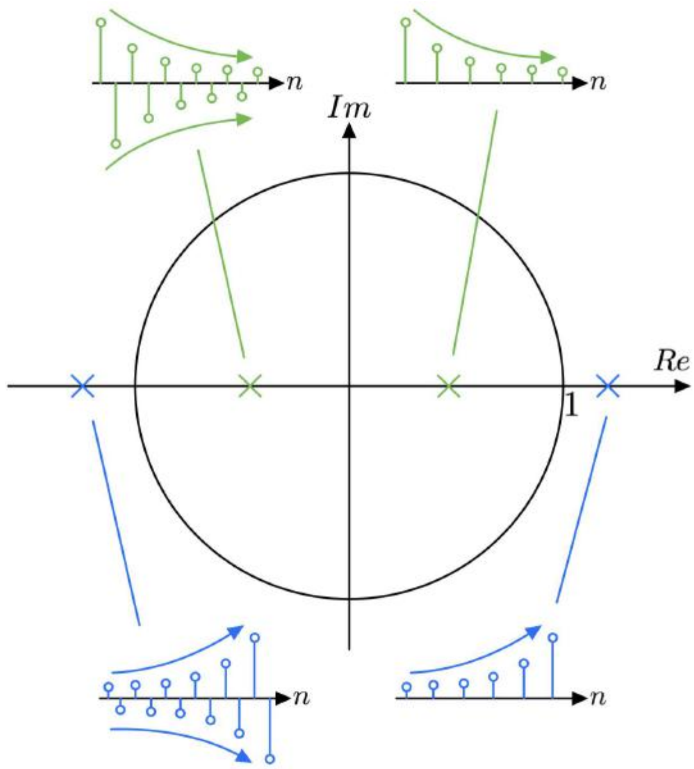
B) Given auxiliary conditions (you must use yh/yp method)
   1. Find the N roots of the characteristic polynomial
   2. Using the roots, find the general form of the homogeneous solution
   3. Find the particular solution
   4. Plug the values into the DE to find the full particular solution
   5. Combine both the homogeneous solution and the particular solution
   6. Plug in the auxiliary conditions to find the unknown coefficients in the yh part of the solution to get the final answer

Table for particular solutions:

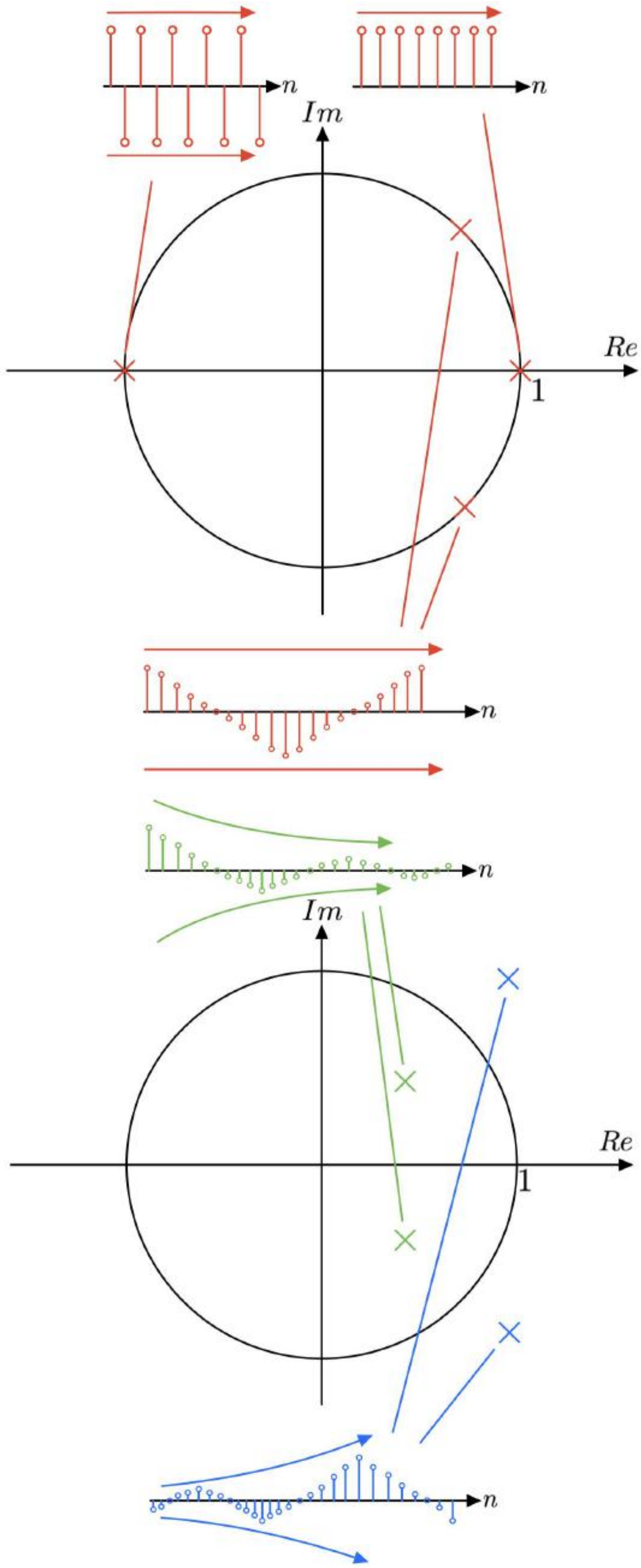
Input form	**Particular solution form**
$x[n] = r^n$ where $r \neq z_m; m = 1, 2, ..., N$	Cr^n
$x[n] = r^n$ where $r = z_m$ for some $m = 1, 2, ..., N$	Cnr^n
$\cos(\beta n + \theta)$	$C\cos(\beta n + \theta)$
$\left(\sum_{i=0}^{m} \alpha_i n^i\right) r^n$	$\left(\sum_{i=0}^{m} C_i n^i\right) r^n$

Modes and Poles

Discrete time systems are stable when their poles lie within the unit circle (these are highlighted in green below). Unstable systems have their poles outside the unit circle (these are highlighted in blue below). If a system's poles lie on the unit circle, the system is said to be metastable (these are highlighted in red below). The poles of a system, z_m, are obtained by solving the characteristic polynomial, as described above. They are values on the complex plane that define the modes of the system. The modes of the system are then given by z_m^n, allowing it to repeat indefinitely (recall that, in the case of discrete time systems, n is the time variable). The modes define the behaviour of the system. For example, if a pole is less than one (within the unit circle), as n grows, the value of the function (given by the modes) will tend toward zero (in other words, it will be stable). On the following diagram, the X's represent the poles, while the graphs surrounding the unit circle represent the modes.



n
n
Im
Re
1
n
n



n
n
Im
Re
1
n
n
Im
Re
1
n

Eigenfunctions
An eigenfunction is a family of special functions such that when it is passed as an input to a given system, the output is that same function **multiplied by a constant** known as an eigenvalue. The relation below illustrates this point:

Input = Eigenfunction
→ Output = Eigenvalue X Eigenfunction

Example:

Consider the following system: $y\left(t\right) = \frac{dx(t)}{dt}$

We can find that the function e^{st} is an eigenfunction of the system:
$y(t) = \frac{d}{dt}e^{st} = s \times e^{st}$ where the constant s is the eigenvalue
Thus, given an input of $x(t) = \cos\left(3t\right)$, one can find the output using the eigenfunction method:
$x(t) = 2\cos\left(3t\right) = e^{3jt} + e^{-3jt}$ (Splitting up the input using Euler's formula)
Now that the input is in the form of eigenfunctions, we find the eigenvalues (a = 3j and a = -3j) and write the output as the following:
$y(t) = 3je^{3jt} - 3je^{-3jt}$

To verify that the output is the same as the derivative of $2\cos\left(3t\right)$, we multiply the numerator and denominator of y(t) by 2j:

$$y(t) = \frac{2j \times 3je^{3jt} - 2j \times 3je^{-3jt}}{2j} = 2\frac{-3e^{3jt} + 3je^{-3jt}}{2j} = -2\sin\left(3t\right)$$

Transfer Functions
In general, each system has its unique transfer function which contains information on how the system maps the input to the output:

For CT systems: $H(s) = \int_{-\infty}^{\infty} h(\tau)e^{s\tau}\,d\tau$

For DT systems: $H(z) = \sum_{m=-\infty}^{\infty} h[m]z^{-m}$

Once the transfer function of a system is found, the output can be mapped from the input using the following relations:

$y(t) = H(s)\,x(t)$ where $x(t) = e^{st}$
$y[n] = H(z)x[n]$ where $x[n] = z^{m}$

Thus, the general approach to find the output y from the input x needs the following steps:

1- Write the input as a sum of the eigenfunctions ($e^{st}\ for\ CT\ and\ z^{-m}\ for\ DT$)
2- Find the transfer function of the system
3- Plug in the value of s into the transfer function to give the eigenvalue
4- Multiply the eigenfunction by the eigenvalue to get the output

The first step is usually the most challenging. Many different techniques can be used to write the input in the form of eigenfunctions (Fourier Series, Fourier Transform, Z-Transform, Laplace Transform).

Deriving Transfer Functions from ODE's and DE's

Given an ODE in the form: $ay''(t) + by'(t) + cy(t) = dx''(t) + ex'(t) + fx(t)$
The transfer function of the system can be written as the following:
$H(s) = \frac{ds^2+es+f}{as^2+bs+c}$

For DT systems, given
$ay[n] + by[n-1] + cy[n-2] = dx[n] + ex[n-1] + fx[n-2]$
$H(z) = \frac{d+ez^{-1}+fz^{-2}}{a+bz^{-1}+cz^{-2}}$

Transfer Functions and Sinusoidal Input

When the input to the system is a sinusoid (sine or cosine), there is no need to convert the signal to a sum of eigenfunctions (step 1). Once the transfer function is found, the output can be simply calculated as follows:

For CT systems:
$Given\ x(t) = A\cos(\omega t + \phi)$
$y(t) = |H(j\omega)|A\cos(\omega t + \phi + \angle H(j\omega))$
$Where:$
$H(s) = H(j\omega)$
$|H(j\omega)|\ is\ the\ magnitude\ of\ the\ transfer\ function$
$\angle H(jw)\ is\ the\ phase\ of\ the\ transfer\ function$

For DT systems:
$Given\ x[n] = A\cos(\omega n + \phi)$
$y[n] = |H(z)|A\cos(\omega n + \phi + \angle H(z))$

Complex Exponential Signals

Periodic signals (for example, $x(t) = \cos(t)$) can be expressed as functions of periodic exponential signals with imaginary exponents through the Fourier series. Aperiodic signals with finite energy, meaning that they don't go to infinity (for example, $x(t) = 5t - 3$), can also be expressed as functions of periodic exponential signals with imaginary exponents. However, in such cases, the Fourier *transform* is used. This will be discussed in upcoming lectures.

The smallest real, positive value of T for which $x(t) = x(t + T)$ holds is the fundamental period of the signal x(t). For periodic signals such as $\cos(\omega_0 t)$, $\sin(\omega_0 t)$, and $e^{j\omega_0 t}$, the fundamental period is $T = \frac{2\pi}{\omega_0}$. Here, ω_0 is the fundamental frequency.

The Fourier Series

When signals are represented using a series of complex exponentials with imaginary exponents as follows, this is called the *Fourier series representation* of the signal.

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t} = \sum_{-\infty}^{\infty} a_k e^{jk\frac{2\pi}{T}t}$$

The "k" in the equation above is the harmonic. When k=0, that component of x(t) is constant. For $k = \pm 1$, we refer to the "first harmonic" component of the signal (also known as the fundamental harmonic). The pattern continues with increasing absolute values of k.

Sinusoidal functions can be quickly converted to their Fourier series equivalents by using Euler's relation. For example, given the signal $x(t) = \cos(3t)$, we can use Euler's relation, that is, $e^{jt} = \cos(t) + j\sin(t)$, to obtain the Fourier series representation of the signal: $x(t) = \frac{1}{2}\left(e^{j3t} + e^{-j3t}\right)$. In this case, the fundamental frequency is 3. We can then start looking at defining the coefficients "a_k" of the Fourier series. As we can see, in this example, there is only a term involving the fundamental frequency. This will be the term associated with $k = \pm 1$. Looking at the coefficient of the pair of exponential terms in parentheses, we can see that $a_{\pm 1} = \frac{1}{2}$. Finally, we can say that $a_{k \neq \pm 1} = 0$.

In the case of more complex signals (for example, see below), it is important to keep track of the fundamental frequency. Note that, here, all other signal frequencies are multiples of 2π, making it the fundamental frequency.

$$x(t) = 1 + \frac{1}{2}\cos(2\pi t) + \cos(4\pi t) + \frac{2}{3}\cos(6\pi t)$$

However, consider the case where, instead, we make the following very small change:

$$x(t) = 1 + \frac{1}{2}\cos(2\pi t) + \cos(4\pi t) + \frac{2}{3}\cos(5\pi t)$$

Suddenly, the fundamental frequency becomes π!

Fourier Coefficients as Complex Numbers

The Fourier coefficients a_k are complex numbers. They can be written in both cartesian and polar forms. Very often, we tend to graph the magnitude of the coefficient alone and the phase alone.

We notice that for any **real signal**, opposite Fourier coefficients will always be complex conjugates, which implies the following:

$$a_k = a^{\star}_{-k} \rightarrow |a_k| = |a_{-k}| \; \angle a_k = -\angle a_{-k}$$

$$\Re\{a_k\} = \Re\{a_{-k}\} \; Im\{a_k\} = -Im\{a_{-k}\}$$

If the signal is **real even**, then the imaginary part of all Fourier coefficients vanishes to zero and the coefficients are all **pure real**, as a result:

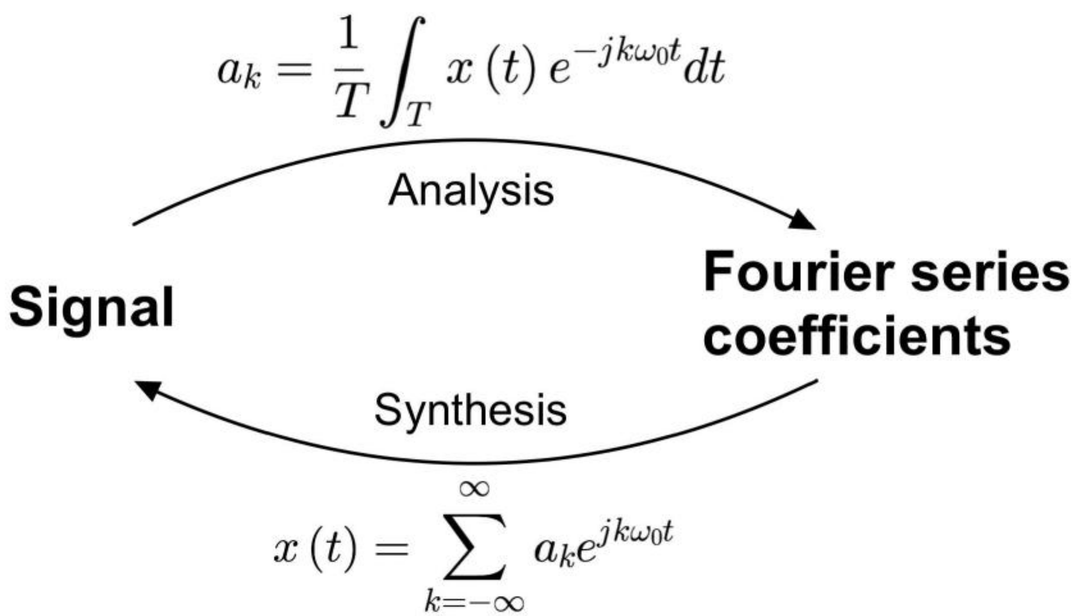
$$\angle a_k = \angle a_{-k} = 0$$

$$Im\{a_k\} = Im\{a_{-k}\} = 0$$

$$\rightarrow a_k = a_{-k}$$

If the signal is **real odd**, then the real part of all Fourier coefficients becomes zero and the coefficients are either **pure imaginary** or zero (if the imaginary part is already zero):

$$a_k = -a_{-k} = Im\{a_k\}$$



The above two equations are the most important equations of the lecture. The analysis equation allows us to take the signal x(t) and obtain the Fourier series coefficients. Conversely, the synthesis equation allows us to take the Fourier series coefficients a_k and obtain the signal. Recall that here, $\omega_0 = \frac{2\pi}{T}$.

Properties of the CTFS

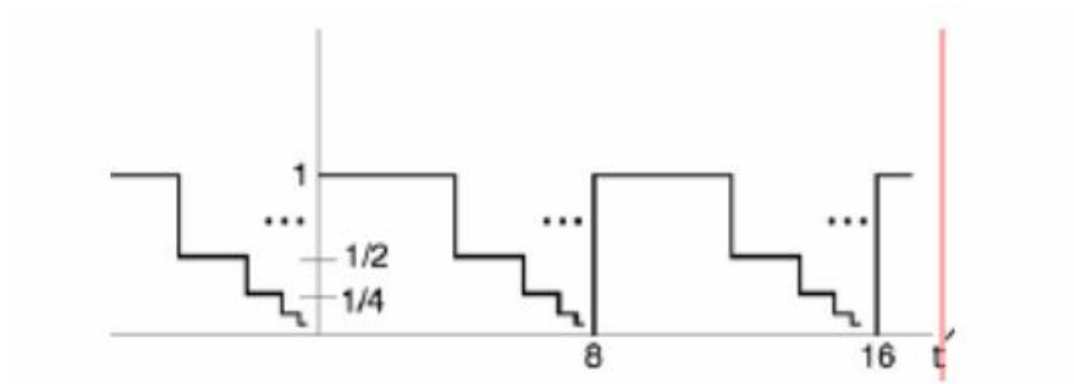
PROPERTY	WITH MATH	WITH WORDS
Linearity	$x(t) \leftrightarrow a_k, y(t) \leftrightarrow b_k \Rightarrow \alpha x(t) + \beta y(t) \leftrightarrow \alpha a_k + \beta b_k$	If x(t) and y(t) have Fourier series, you can combine them and then find a Fourier series *or* find each of their Fourier series and then combine those series.
Time Reversal	$x(t) \leftrightarrow a_k \Rightarrow x(-t) \leftrightarrow a_{-k}$ If x(t) even, even coefficients: $a_k = a_{-k}$ If x(t) odd, odd coefficients: $a_k = -a_{-k}$	Reversing time reverses the subscript sign.
Time Shifting	$x(t - t_0) \leftrightarrow a_k e^{-jkw_0 t_0}$ with $w_0 = \dfrac{2\pi}{T}$	A time shift in the time domain is a phase shift in the frequency domain. There will be no change in magnitude.
Conjugate Symmetry	x(t) real: $a_k^* = a_{-k}$ x(t) real, even: $a_k = a_{-k} = a_k^*$ only real/even coefficients x(t) real, odd: only imaginary/odd coefficients	In all cases, at k and -k, there will be the same REAL parts to the coeffients, but the PHASES will be inversed.
Time Scaling	$x(\beta t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\beta\omega_0 t}$	N/A
Multiplication	If $x(t) \leftrightarrow a_k$ and $y(t) \leftrightarrow b_k$ then, $x(t)y(t) \leftrightarrow h_k = \sum_{l=-\infty}^{\infty} a_l b_{k-l}$	Multiplying two periodic signals with the SAME PERIOD in the time is equivalent to discrete time convolution in the frequency domain.
Parseval's Theorem	$\frac{1}{T}\int_T \lvert x(t)\rvert^2 dt = \sum_{k=-\infty}^{\infty} \lvert a_k \rvert^2$	The average power in one period is equal to the sum of the power of each harmonic component
Periodic Convolution	Given $x(t) \leftrightarrow a_k$ and $y(t) \leftrightarrow b_k$, looking for $z(t) \leftrightarrow c_k$ then, $c_k = Ta_k b_k$ where T is the period	The period T must be common to all three signals.

Convergence of a Fourier Series

Convergence implies that we can represent a signal using a Fourier series. For a signal to satisfy this, it must be periodic <u>and</u> one of the following:

1. Continuous (e.g. cosine signal)
2. Discontinuous but with finite energy over each period (e.g. a square wave):
   $\int_T |x(t)|^2 dt < \infty$

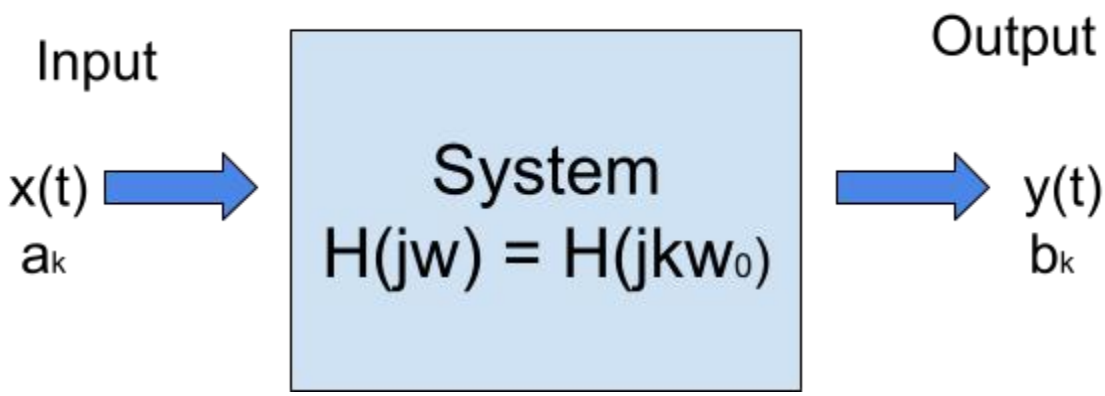
3. Satisfies All 3 Dirichlet Conditions:
   a. Must be absolutely integrable
      $\int_T |x(t)| dt < \infty$
      Counter example: $x(t) = \frac{1}{t}$
   b. In any one of the periods there must be a finite number of discontinuities
      Counter example:



   c. In any one of the periods there must be a finite number of oscillations:
      Counter example: $x(t) = \sin\left(\frac{1}{t}\right)$

Fourier Series and Systems

Suppose an input signal for a certain system x(t) has Fourier Series Coefficients a_k and the output signal y(t) has fourier series coefficients b_k. The system's transfer function $H(j\omega)$ is used to map from a_k to b_k:



$b_k = H(jkw_0) \times a_k$

Knowing that the fourier coefficients and the transfer function are complex numbers,

$|b_k| = |H(jk\omega_0)| \times |a_k|$

$\angle b_k = \angle H(jk\omega_0) + \angle a_k$

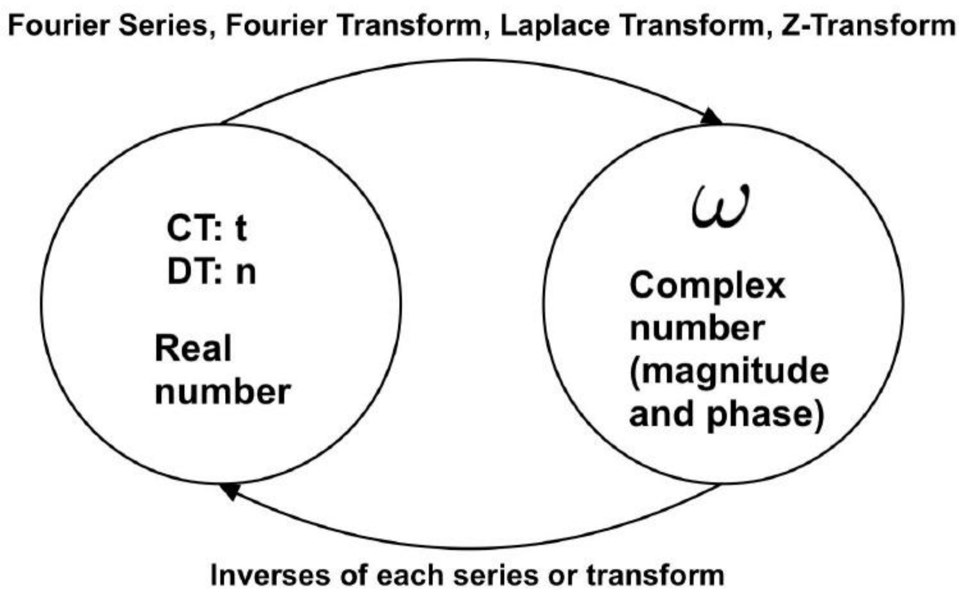
Reminder: The transfer function can be found from the impulse response h(t) using the following formula:

$H(jw) = \int_{-\infty}^{\infty} h(\tau)\, e^{-jk\tau}\, d\tau$

We will find out later that the transfer function is the **Fourier Transform** of the impulse response.

Time and Frequency Domains:

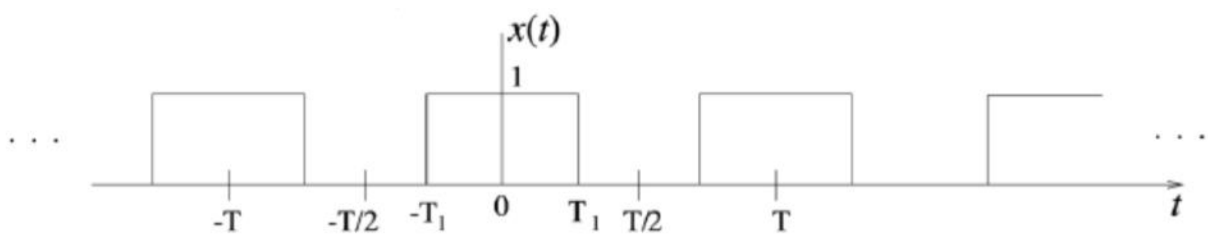
In BIEN 350, we examine signals in 2 different worlds. The original signal is said to be in the *time domain*. It is a function of time, which is plotted on the horizontal axis. In other words, the signal depicts an evolution through time. It is a depiction of the frequencies of the signal. In the case of a periodic signal, the frequencies are visually noticeable. For example, consider a signal with a period of 0.5 seconds. It completes two full oscillations per second (two *Hertz*, Hz). This signal would be depicted in the frequency domain by two impulse functions mirrored around the vertical axis at -2Hz and 2Hz. So far, we've seen that Fourier Series give rise to the Fourier Coefficients, which is a discrete value representation of a signal in the frequency domain (works only for periodic CT signals). Other tools that are used to move between those domains are Fourier Transform, Laplace Transform, Z-transform and their inverses.



Note: The inverse of the Fourier Series is the Synthesis Equation

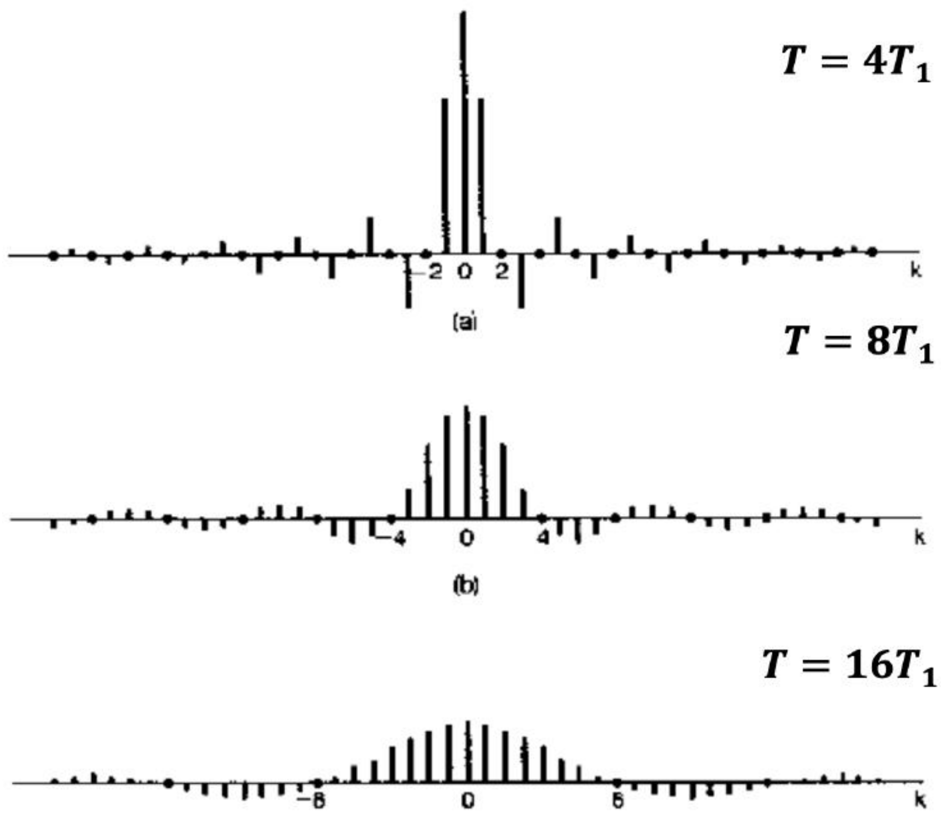
From Fourier Series to Fourier Transform

Fourier Transforms are used mainly for aperiodic signals. To better understand the difference between Fourier Series and Fourier Transforms, one must examine periodic signals. Consider the signal below and its Fourier Series at the lowest fundamental period (1 square cycle):

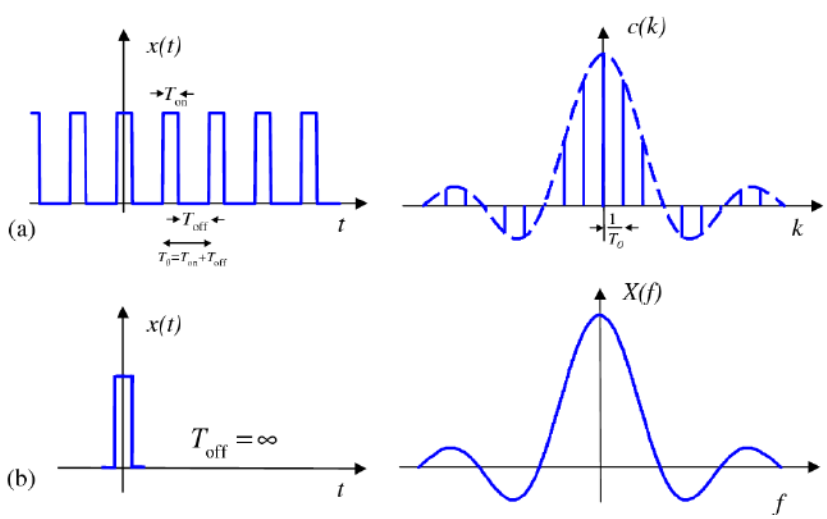


As the fundamental period increases (2 squares, 3 squares, 4 squares…), we observe 2 things:

1) The resolution of the frequency domain increases
2) The amplitude of the individual spikes decreases to accommodate their increasing frequency.



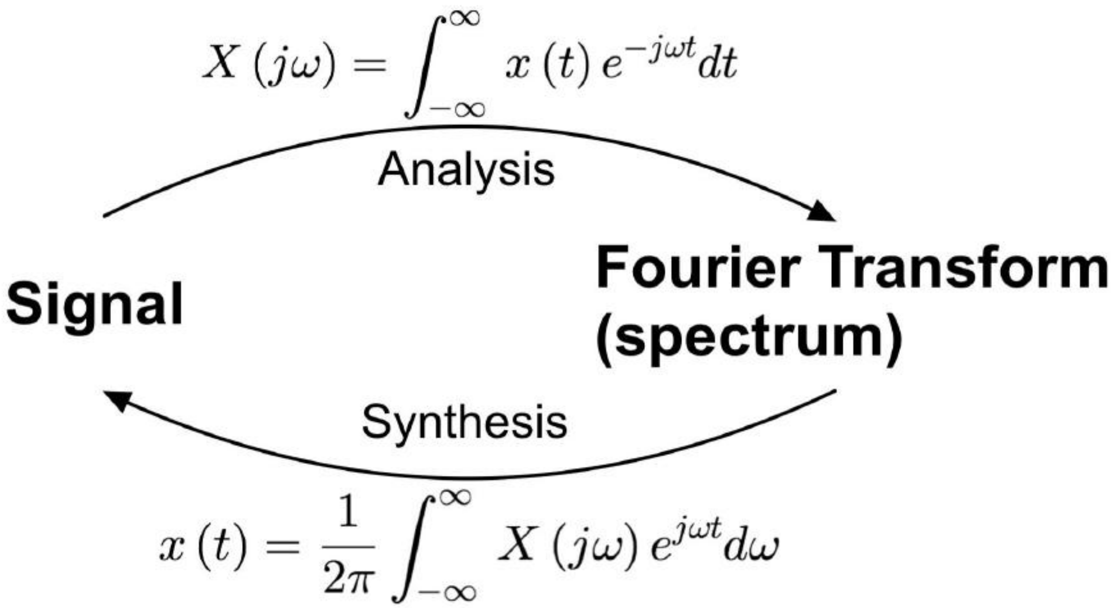
As the fundamental period approaches infinity, the resolution of the frequency domain becomes even more fine until they become CONTINUOUS. A Fourier Transform is thus the application of a Fourier Series on an **infinite fundamental period** which gives rise to a **continuous** Frequency domain as opposed to the **discrete** Frequency domain of the Fourier Series.



https://www.semanticscholar.org/paper/Power-Spectrum-and-Correlation-9.1-Power-Spectrum/4311b940deedd25ee1654e855885a9c49efc1bd8/figure/2

From Signal to Fourier Transform

The Fourier transform of the signal $x(t)$ is given by $F\{x(t)\} = X(j\omega)$. Conversely, the signal can be defined as the inverse of the Fourier Transform: $x(t) = F^{-1}\{X(j\omega)\}$. Overall, the Fourier Transform pair is written as $x(t) \underset{\leftrightarrow}{F} X(j\omega)$. We can transition from the signal to the Fourier Transform of the signal by using the analysis equation (shown below). Conversely, we can transition from the Fourier Transform of a signal to the signal itself by using the synthesis equation (shown below).



Fourier Transform Convergence

The convergence conditions are similar to those of the Fourier Series, albeit that not any continuous signal can have a Fourier Transform:

1. The signal has finite energy over the entire time domain:
   $\int_{-\infty}^{\infty} |x(t)|^2 dt < \infty$

2. The signal satisfies All 3 Dirichlet Conditions:
   a. Must be absolutely integrable
   b. In any one of the periods there must be a finite number of discontinuities
   c. In any one of the periods there must be a finite number of oscillations:

General Steps to Solving a Fourier Transform Problem

Given a certain signal, we would like to find the Fourier Transform:
1- Check For Fourier Pairs that already exist in the table

2- If step 1 fails, check for transformations that might lead to one of the fourier pairs in the table (using properties or FT)
3- If step 2 fails, use the analysis equation
4- At the end, we obtain a complex number. Often, the question will ask you to find/graph the amplitude and phase of this complex number. If the complex number involves several terms, you can find magnitude and phase for each then use the basic operations of complex numbers to merge them together and get the global magnitude and phase.

Properties of Fourier Transform

PROPERTY	TIME DOMAIN	FREQUENCY DOMAIN	EXPLANATION
Linearity	$\alpha x(t) + \beta y(t)$	$\alpha X(j\omega) + \beta Y(j\omega)$	N/A
Time Shifting	$x(t - t_0)$	$e^{-jwt_0}X(j\omega)$	Time shift in time domain = Phase shift in frequency domain.
Time Scaling	$x(\alpha t)$	$\frac{1}{\|\alpha\|}X(\frac{j\omega}{\alpha})$	Compression in time domain = Expansion in frequency domain. (+vice versa)
Time Inversion	$x(-t)$	$X(-j\omega)$	Inversion in one domain is inversion in the other, via the Time Scaling property.
Conjugate Symmetry	1. $x^*(t)$ 2. **Real** x(t) 3. **Real** $x(t) = x_e(t) + x_o(t)$ A. $x_e(t)$ B. $x_o(t)$ Notes: - for x(t) real/even: X(jw) real/even - for x(t) real/odd: X(jw) imaginary/odd	1. $X^*(-jw)$ 2. $X(-jw) = X^*(jw)$, $\|X(-jw)\| = \|X(jw)\|$, $\angle X(-jw) = -\angle X(jw)$ 3. A. $Re\{X(j\omega)\}$ B. $jIm\{X(j\omega)\}$	1. N/A 2. Real x(t) = even FT magnitude, odd FT phase 3. N/A
Differentiation	$\frac{dx(t)}{dt}$	$j\omega X(j\omega)$	Amplification of higher frequencies.
Integration	$\int_{-\infty}^{t} x(\tau)d\tau$	$\frac{X(j\omega)}{j\omega} + \pi X(j0)\delta(\omega)$	Amplification of lower frequencies.
Duality	1. $x(t)$ 2. $X(t)$ 3. $e^{j\omega_0 t}x(t)$ 4. $-jtx(t)$	1. $X(j\omega)$ 2. $2\pi x(-\omega)$ 3. $X(j(\omega - \omega_0))$ 4. $\frac{dX(j\omega)}{d\omega}$	1. N/A 2. N/A 3. Shift in frequency domain = Linear phase shift w.r.t. t in time domain. 4. N/A
Parseval's Theorem	$\int_{-\infty}^{\infty} \|x(t)\|^2 dt = \ldots$	$\ldots = \frac{1}{2\pi}\int_{-\infty}^{\infty} \|X(j\omega)\|^2 d\omega$	Energies are equivalent in both domains.
Convolution	1. $x(t) \circledast y(t)$ 2. $x(t) \cdot y(t)$	1. $X(j\omega) \cdot Y(j\omega)$ 2. $\frac{1}{2\pi}X(j\omega) \circledast Y(j\omega)$	N/A

Fourier Transform Pairs

	$f(t)$	$F(\omega)$	
1	$e^{-at}u(t)$	$\dfrac{1}{a+j\omega}$	$a>0$
2	$e^{at}u(-t)$	$\dfrac{1}{a-j\omega}$	$a>0$
3	$e^{-a\|t\|}$	$\dfrac{2a}{a^2+\omega^2}$	$a>0$
4	$te^{-at}u(t)$	$\dfrac{1}{(a+j\omega)^2}$	$a>0$
5	$t^n e^{-at}u(t)$	$\dfrac{n!}{(a+j\omega)^{n+1}}$	$a>0$
6	$\delta(t)$	1	
7	1	$2\pi\delta(\omega)$	
8	$e^{j\omega_0 t}$	$2\pi\delta(\omega-\omega_0)$	
9	$\cos\omega_0 t$	$\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$	
10	$\sin\omega_0 t$	$j\pi[\delta(\omega+\omega_0)-\delta(\omega-\omega_0)]$	
11	$u(t)$	$\pi\delta(\omega)+\frac{1}{j\omega}$	
12	$\operatorname{sgn} t$	$\frac{2}{j\omega}$	
13	$\cos\omega_0 t\, u(t)$	$\frac{\pi}{2}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]+\frac{j\omega}{\omega_0^2-\omega^2}$	
14	$\sin\omega_0 t\, u(t)$	$\frac{\pi}{2j}[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)]+\frac{\omega_0}{\omega_0^2-\omega^2}$	
15	$e^{-at}\sin\omega_0 t\, u(t)$	$\frac{\omega_0}{(a+j\omega)^2+\omega_0^2}$	$a>0$
16	$e^{-at}\cos\omega_0 t\, u(t)$	$\frac{a+j\omega}{(a+j\omega)^2+\omega_0^2}$	$a>0$
17	$\operatorname{rect}\left(\frac{t}{\tau}\right)$	$\tau\operatorname{sinc}\left(\frac{\omega\tau}{2}\right)$	
18	$\frac{W}{\pi}\operatorname{sinc}(Wt)$	$\operatorname{rect}\left(\frac{\omega}{2W}\right)$	
19	$\Delta\left(\frac{t}{\tau}\right)$	$\frac{\tau}{2}\operatorname{sinc}^2\left(\frac{\omega\tau}{4}\right)$	
20	$\frac{W}{2\pi}\operatorname{sinc}^2\left(\frac{Wt}{2}\right)$	$\Delta\left(\frac{\omega}{2W}\right)$	
21	$\displaystyle\sum_{n=-\infty}^{\infty}\delta(t-nT)$	$\displaystyle\omega_0\sum_{n=-\infty}^{\infty}\delta(\omega-n\omega_0)$	$\omega_0=\frac{2\pi}{T}$
22	$e^{-t^2/2\sigma^2}$	$\sigma\sqrt{2\pi}e^{-\sigma^2\omega^2/2}$	

Defining Frequency Response

Simply put, the frequency response $H(j\omega)$ is the impulse response $h(t)$ in the frequency domain. This allows us to map the input in the frequency domain using the following formula:

$$Y(j\omega) \;=\; H(j\omega)X(j\omega)$$

Some of the methods used to find Frequency response:

1. Taking the fourier transform of the impulse response
2. Extracting it from an ODE
3. Complex impedances (given a circuit)
4. If the transfer function H(s) is already given, we can find the frequency response by simply replacing s with $j\omega$ to get the frequency response
5. Experimental/Practical Method by passing into the system different sinusoidal inputs of different frequencies.

Being an imaginary number, the frequency response has a magnitude and a phase, which can be plotted against the ω axis to show the response of the system to different frequencies.

For a frequency response to exist, a system must be BIBO stable, meaning that $\int_{-\infty}^{\infty}|h(t)|\,dt \;<\infty$

Partial Fraction Decomposition

1. Factorize the denominator
2. Write the denominator as a sum of fractions where each fraction has its numerator as an unknown constant(s) and the denominator as one of the terms.The table below shows the decomposition general form for each type of factor in the denominator:

Factor in denominator	Term in partial fraction decomposition
$ax+b$	$\dfrac{A}{ax+b}$
$(ax+b)^k$	$\dfrac{A_1}{ax+b}+\dfrac{A_2}{(ax+b)^2}+\cdots+\dfrac{A_k}{(ax+b)^k},\ k=1,2,3,\ldots$
ax^2+bx+c	$\dfrac{Ax+B}{ax^2+bx+c}$
$(ax^2+bx+c)^k$	$\dfrac{A_1x+B_1}{ax^2+bx+c}+\dfrac{A_2x+B_2}{(ax^2+bx+c)^2}+\cdots+\dfrac{A_kx+B_k}{(ax^2+bx+c)^k},\ k=1,2,3,\ldots$

https://tutorial.math.lamar.edu/classes/calcii/partialfractions.aspx

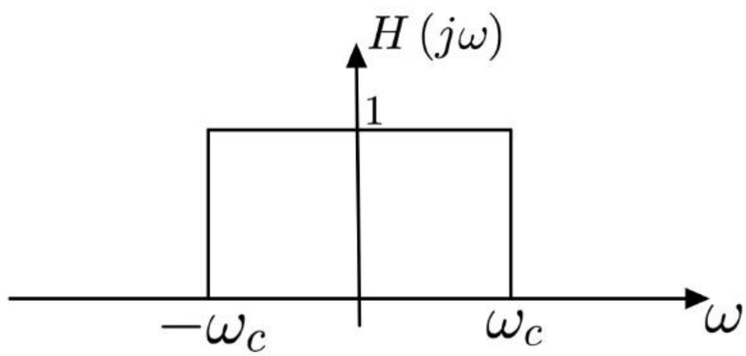
3. Using common denominator, combine all the terms together into 1 term

4. Either by grouping like terms in the numerator or by plugging in values for the variable x, establish a system of equations that allows us to find the values of the unknowns A,B,C...

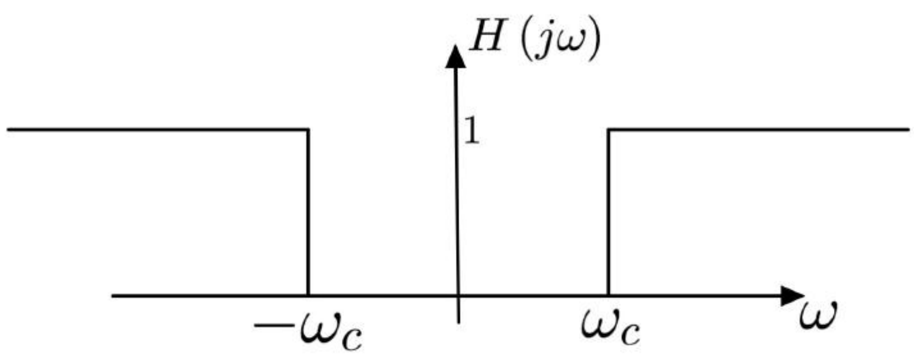
Filtering and Ideal Filters

A filter is a system that takes the frequencies that make up the input signal and either amplifies, attenuates, conserves, or deletes each frequency. The ideal filter is a filter where certain frequencies are kept the same while all other frequencies are removed. Below are the 4 types of ideal filters:

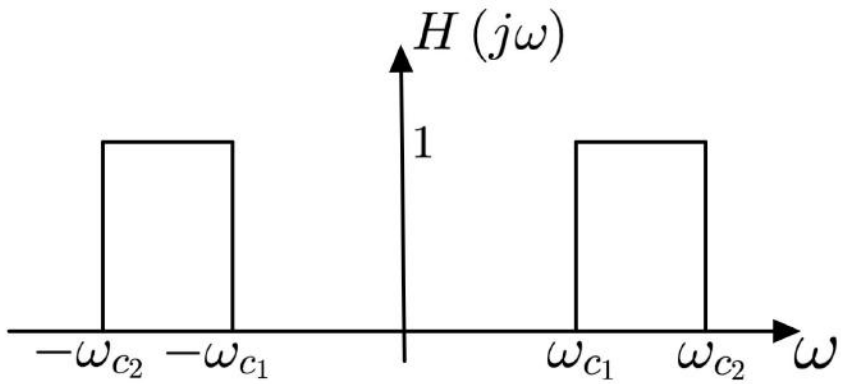
A) *Low Pass Filters: Only frequencies lesser in absolute value than the cutoff frequency are retained.*



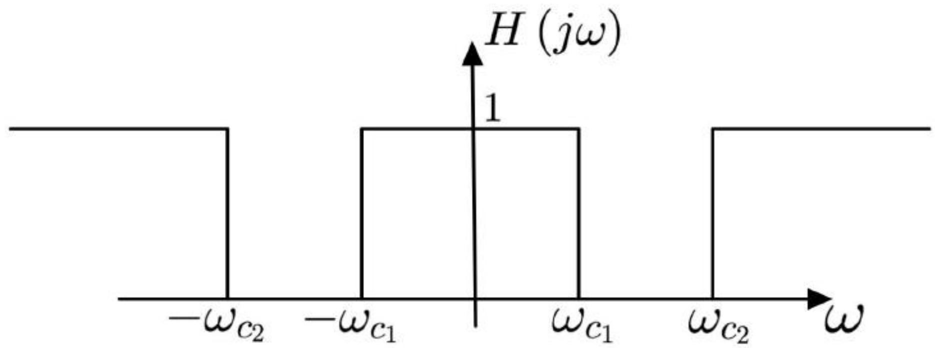
B) *High Pass Filters: Only frequencies greater in absolute value than the cutoff frequency are retained.*



C) *Band Pass Filters: All frequencies between the two cutoff frequencies are retained.*



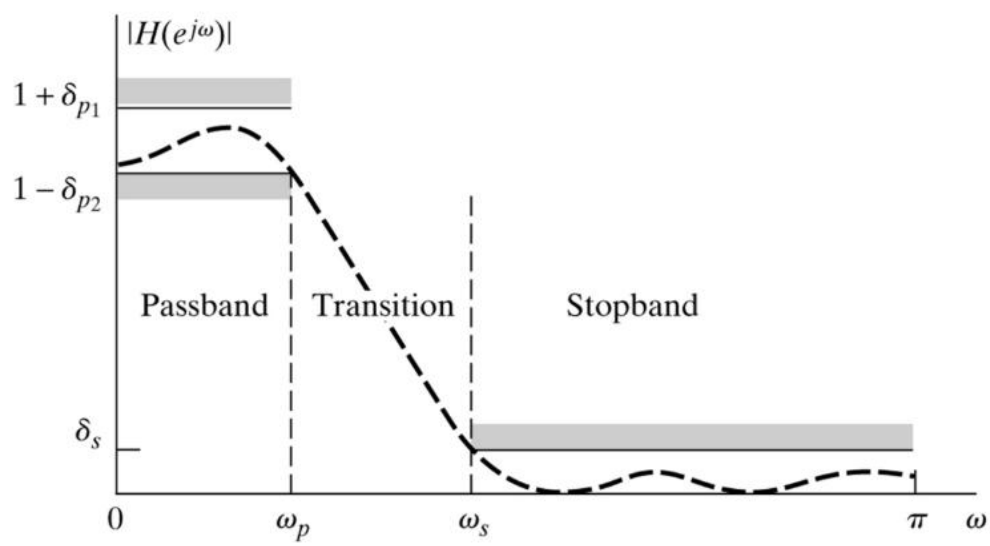
D) *Band Stop Filters: All frequencies are retained except those between the two cutoff frequencies.*



Non-Ideal Filters: In the case of non-ideal filters, the following phenomena can be found:

1- Cutoff is non-discrete

2- Oscillations along the frequency axis in both high and low modes



Examples of non-ideal filters: Butterworth and Chebyshev Filters

ODEs and Frequency Response

Recall from before that given an ODE in the form:

$ay''(t) + by'(t) + cy(t) = dx''(t) + ex'(t) + fx(t)$

The transfer function of the system can be written as the following:

$$H(s) = \frac{ds^2+es+f}{as^2+bs+c}$$

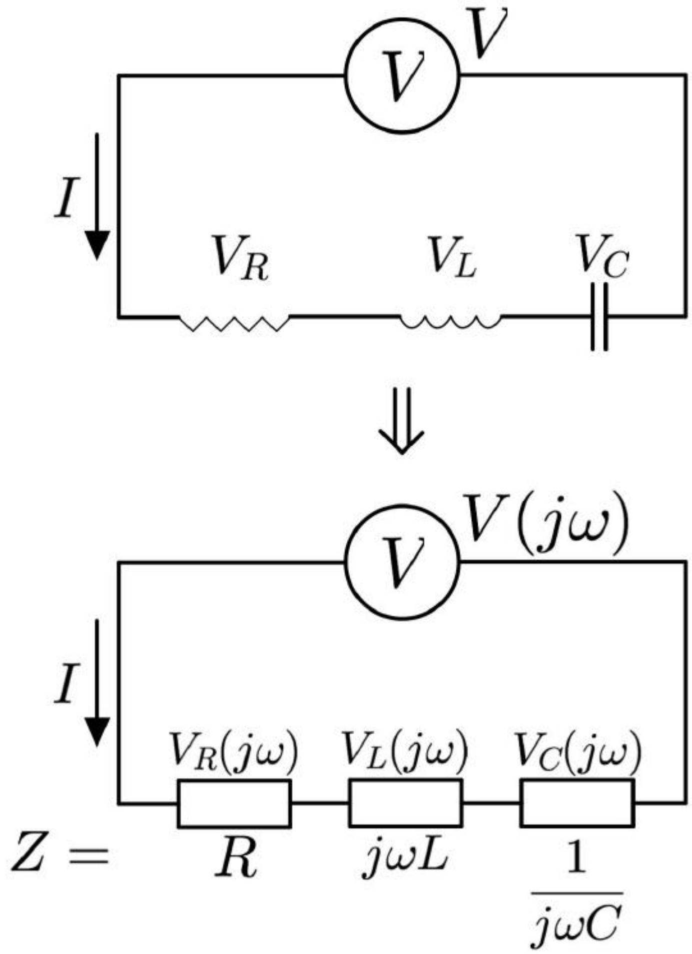
Finding the frequency response is very similar, we just replace the s with jw:

$$H(j\omega) = \frac{d(j\omega)^2+ej\omega+f}{a(j\omega)^2+bj\omega+c} = \frac{-d\omega^2+ej\omega+f}{-a\omega^2+bj\omega+c}$$

Complex Impedances

The method of complex impedances is used to solve circuit problems that involve frequency response. This involves using impedance (Z), which is the resistance to AC voltage (basically resistance in the frequency domain). Below is a general approach used to solve complex impedance circuit problems:

1. Transform the components of the circuit into 'special' impedance resistors. The impedance of these imaginary resistors can be calculated based on the nature of each component:
   Resistor: $Z_R(j\omega) = R$
   Capacitor: $Z_c(j\omega) = \frac{1}{j\omega C}$
   Inductor: $Z_L(j\omega) = j\omega L$
2. Write the input/output relation using one of Kirchoff's laws. Denote all voltages and currents in the frequency domain (e.g. use $V(j\omega)$ and $I(j\omega)$ instead of $V(t)$ and $i(t)$)
3. Treating the impedances as resistances, we can do any of the following tricks to isolate input and output:
   a. Ohm's law of a component: $V(j\omega) = Z(j\omega) \times I(j\omega)$
   b. Equivalent impedance of components in series: $Z_{eq}(j\omega) = Z_1(j\omega) + Z_2(j\omega)$
   c. Equivalent impedance of parallel components: $\frac{1}{Z_{eq}(j\omega)} = \frac{1}{Z_1(j\omega)} + \frac{1}{Z_2(j\omega)}$
   d. Voltage Divider Trick: When there are components that are **only in series**, this special relation arises between the total voltage and the voltage of one of the components:
   $$V_{total}(j\omega) = \frac{Z_{component}(j\omega)}{Z_{total}(j\omega)} V_{component}(j\omega)$$
4. Reduce the expression to the form $Y(j\omega) = H(j\omega)\,X(j\omega)$ to find $H(j\omega)$


V
V
I
V_R
V_L
V_C
⇓
$V(j\omega)$
V
I
$V_R(j\omega)$
$V_L(j\omega)$
$V_C(j\omega)$
$Z =$
R
$j\omega L$
$\frac{1}{j\omega C}$

Frequency Domain for DT signals

Compared to CT signals, there are some differences that we observe when dealing with the frequency domain of DT signals:

1. The frequency domain representation of a signal x[n] is $X\left(e^{j\omega}\right)$ instead of $X(j\omega)$ (which is used for CT signals).
2. For CT signals, if we vary the frequencies ω in $X(j\omega)$, we can always get different values for $X(j\omega)$. However, in DT signals, if we vary the value of ω in $X\left(e^{j\omega}\right)$, the frequency domain repeats itself at every frequency multiple of 2π. For example: $X\left(e^{3\pi j}\right) = X\left(e^{5\pi j}\right) = X\left(e^{\pi j}\right)$. Hence, the frequency domain of a DT signal has a period of 2π.
3. The transfer function of a CT system is H(s), whereas the transfer function of a DT system is H(z).
   Spoiler Alert: You will see later on in the course that $s = \sigma + j\omega$ and $z = re^{jw}$. For this chapter and the ones before it, we consider only the case where $\sigma = 0$ and $r = 1$. Hence, the transfer functions are $H(jw)$ for CT signals and $H\left(e^{jw}\right)$ for DT signals.

The Half-Angle Hack

Very often in BIEN350, when simplifying our answer, we run into the following type of expression:

$Answer = 1 + e^{j\alpha}$ where α is any value

The half-angle hack reduces this expression to the following:

$$1 + e^{j\alpha} = e^{j\frac{\alpha}{2}}\left(e^{j\frac{-\alpha}{2}} + e^{j\frac{\alpha}{2}}\right)$$

Why use this trick? Right now, we have the answer as a product of 2 terms. One of those terms can be reduced to a pure real number and the other can be reduced to a complex number of magnitude 1. This is extremely helpful in finding the magnitude and phase of our answer.

$$e^{j\frac{\alpha}{2}}\left(e^{j\frac{-\alpha}{2}} + e^{j\frac{\alpha}{2}}\right) = e^{j\frac{\alpha}{2}}\left(2\cos\left(\frac{\alpha}{2}\right)\right)$$ (Using Euler's Formula)

$|Answer| = 2\cos\left(\frac{\alpha}{2}\right)$

$\angle Answer = \frac{\alpha}{2}$

A slightly more complicated case in which we can use this hack is the following:

$$Answer = 1 - e^{j\alpha}$$
$$= e^{j\frac{\alpha}{2}}\left(e^{j\frac{-\alpha}{2}} - e^{j\frac{\alpha}{2}}\right)$$
$$= e^{j\frac{\alpha}{2}}\left(2j\,sin\left(\frac{\alpha}{2}\right)\right)$$ (Using Euler's Formula for sine this time)

$$= e^{j\frac{\alpha}{2}}\left(2e^{j\frac{\pi}{2}}\ sin\left(\frac{\alpha}{2}\right)\right)$$ (Substituting for j)

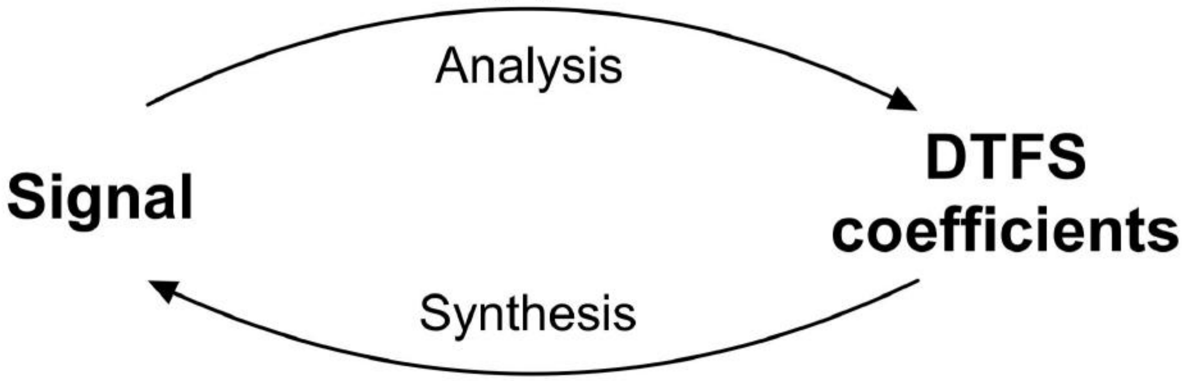
$$= e^{j\frac{\alpha+\pi}{2}}\left(2\,sin\left(\frac{\alpha}{2}\right)\right)$$

$$|Answer| = 2sin\left(\frac{\alpha}{2}\right)$$

$$\angle Answer = \frac{\alpha}{2} + \frac{\pi}{2}$$

DTFS Synthesis and Analysis Equations

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\, e^{-jk\omega_0 n}$$



$$x[n] = \sum_{k=0}^{N-1} a_k e^{jk\omega_0 n}$$

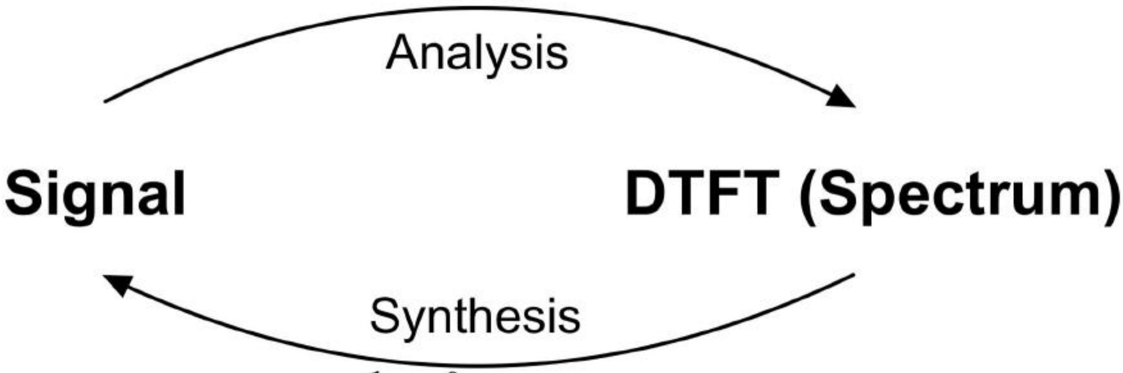
Due to the periodicity of DT signals, DTFS coefficients are periodic with a fundamental period of N as follows: $a_k = a_{k+N}$.

Properties of the DTFS

Property	Periodic sequences $x[n]$, $y[n]$ – period N	DFS coefficients a_k, b_k – period N
Linearity	$Ax[n] + By[n]$	$Aa_k + Bb_k$
Time Shifting	$x[n - n_0]$	$e^{-jk\frac{2\pi}{N}n_0}a_k$
Time Reversal	$x[-n]$	a_{-k}
Periodic convolution	$\sum_r x[r]y[n-r]$	Na_kb_k
Conjugate symmetry		
Real signals	$x[n]$ real	$\begin{cases} a_k = a_{-k}^* \\ \Re\{a_k\} = \Re\{a_{-k}\} \\ \Im\{a_k\} = -\Im\{a_{-k}\} \\ \lvert a_k \rvert = \lvert a_{-k} \rvert \\ \angle a_k = -\angle a_{-k} \end{cases}$
Real and even signals	$x[n]$ real and even	a_k real and even
Real and odd signals	$x[n]$ real and odd	a_k imaginary and odd
Parseval's Theorem	$\frac{1}{N}\sum_{n=0}^{N-1} \lvert x[n] \rvert^2 = \sum_{n=0}^{N-1} \lvert a_k \rvert^2$	

DTFT Synthesis and Analysis Equations

$$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$



$$x[n] = \frac{1}{2\pi}\int_{2\pi} X\left(e^{j\omega}\right) e^{j\omega n} d\omega$$

Properties of the DTFT

PROPERTY	TIME DOMAIN	FREQUENCY DOMAIN	EXPLANATION
Linearity	$\alpha x[n] + \beta y[n]$	$\alpha X(e^{j\omega}) + \beta Y(e^{j\omega})$	N/A
Time Shifting	$x[n - n_d]$	$e^{-j\omega n_d} X(e^{j\omega})$	Time shift in time domain = Phase shift in frequency domain.
Frequency Shifting	$e^{j\omega_0 n} x[n]$	$X(e^{j(\omega-\omega_0)})$	Phase shift in time domain = Frequency shift in frequency domain.
Differentiation (frequency domain)	$n x[n]$	$j \dfrac{dX(e^{j\omega})}{d\omega}$	N/A
Parseval's Theorem	$\sum_{n=-\infty}^{\infty} \lvert x[n] \rvert^2 = \ldots$	$\ldots = \dfrac{1}{2\pi} \int_{-\pi}^{\pi} \lvert X(e^{j\omega}) \rvert^2 d\omega$	Energies are equivalent in both domains.
Convolution	$x[n] \circledast y[n]$	$X(j\omega) \cdot Y(j\omega)$	N/A
Modulation	$x[n] \cdot y[n]$	$\dfrac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta}) Y(e^{j(\omega-\theta)}) d\theta$	Frequency domain convolution over one period of 2π

Sequence $x[n]$	Fourier Transform $X(e^{j\omega})$
1. $x^*[n]$	$X^*(e^{-j\omega})$
2. $x^*[-n]$	$X^*(e^{j\omega})$
3. $\mathcal{R}e\{x[n]\}$	$X_e(e^{j\omega})$ (conjugate-symmetric part of $X(e^{j\omega})$)
4. $j\mathcal{J}m\{x[n]\}$	$X_o(e^{j\omega})$ (conjugate-antisymmetric part of $X(e^{j\omega})$)
5. $x_e[n]$ (conjugate-symmetric part of $x[n]$)	$X_R(e^{j\omega}) = \mathcal{R}e\{X(e^{j\omega})\}$
6. $x_o[n]$ (conjugate-antisymmetric part of $x[n]$)	$jX_I(e^{j\omega}) = j\mathcal{J}m\{X(e^{j\omega})\}$
The following properties apply only when $x[n]$ is real:	
7. Any real $x[n]$	$X(e^{j\omega}) = X^*(e^{-j\omega})$ (Fourier transform is conjugate symmetric)
8. Any real $x[n]$	$X_R(e^{j\omega}) = X_R(e^{-j\omega})$ (real part is even)
9. Any real $x[n]$	$X_I(e^{j\omega}) = -X_I(e^{-j\omega})$ (imaginary part is odd)
10. Any real $x[n]$	$\|X(e^{j\omega})\| = \|X(e^{-j\omega})\|$ (magnitude is even)
11. Any real $x[n]$	$\sphericalangle X(e^{j\omega}) = -\sphericalangle X(e^{-j\omega})$ (phase is odd)
12. $x_e[n]$ (even part of $x[n]$)	$X_R(e^{j\omega})$
13. $x_o[n]$ (odd part of $x[n]$)	$jX_I(e^{j\omega})$

DT Filters

Discrete time filters are similar to those used in continuous time. However, DT filters are periodic with a periodicity of 2π and, therefore, repeat themselves. A lowpass filter, for example, retains the signal between cutoff frequencies around the origin *and* every frequency multiple of 2π. Often, only one period is illustrated, leading to figures almost identical to those shown in lecture 9. The only difference is the vertical axis labelling, which is $H\left(e^{jw}\right)$ in the DT case instead of $H(jw)$ in the CT case.

Magnitude and Phase Response

The magnitude response is the way in which the input signal's magnitude is changed to give the output signal's magnitude. Similarly, the phase response is the way in which the input signal's phase is changed to give the output signal's phase.

Given the input frequencies (spectrum), $X(j\omega)$ or $X\left(e^{j\omega}\right)$, we can use the magnitude response, $|H(j\omega)|$ or $\left|H\left(e^{j\omega}\right)\right|$, and the phase response, $\angle H(j\omega)$ or $\angle H\left(e^{j\omega}\right)$, to compute the magnitude and phase of the output signal's spectrum:
$|Y(j\omega)| = |X(j\omega)||H(j\omega)|$
$\angle Y(j\omega) = \angle X(j\omega) + \angle H(j\omega)$
Note that the magnitudes are multiplied and that the phases are summed.

Linear phase is preferable, as it results in a uniform shifting of the entire signal. Nonlinear phase leads to shifting different frequencies in different ways. This leads to a scrambled output signal.

Group Delay Definition

Group delay is the shift in the **time domain** that a certain frequency of a signal experiences. We denote it mathematically as $\tau(\omega)$. In the frequency domain, the group delay is the derivative of the phase response. It quantifies "how linear" the phase response is; a constant group delay implies linear phase response.

Steps to Obtain the Group Delay

Consider the following system: $y(t) \ = \ 2x(t-3)$

1- Find the Frequency Response of the system
$h(t) \ = \ 2\delta(t-3)$
$H(j\omega) \ = \ \mathcal{F}\{h(t)\} \ = \ 2e^{-3j\omega}$

2- Find the phase of that frequency response:
$\angle H(j\omega) \ = \ -3\omega$

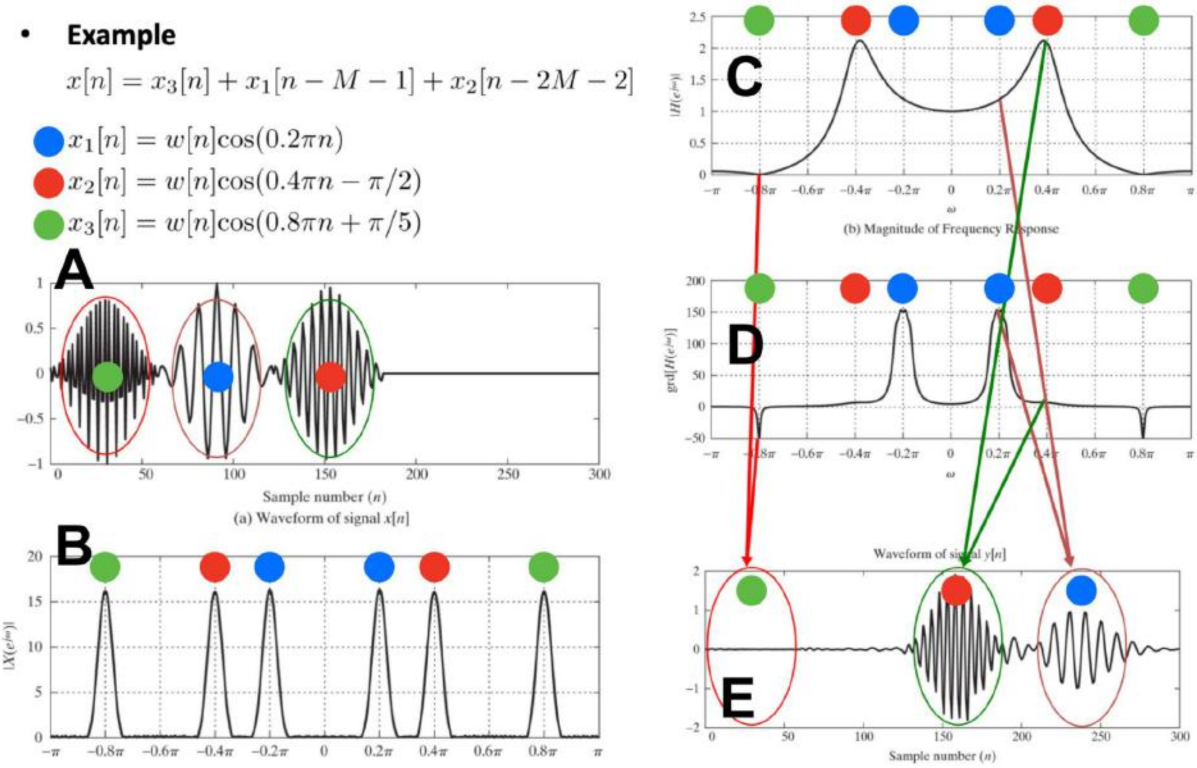
3- Take negative of the derivative of the phase with respect to ω to find the group delay:
$\tau(w) \ = \ -\frac{d}{d\omega}\angle H(j\omega) \ = \ -\frac{d}{d\omega}(-3\omega) \ = \ 3$

This means that for this example:

A) All frequencies in this signal will be amplified by a factor of 2
B) All frequencies in this signal will be shifted in the time-domain by 3 units to the right.

In many other examples, the group delay (and the magnitude response) can differ among different frequencies, which causes different shifts in the time domain.

Important Example: Explained



To simplify the text, henceforth, the plot in the center left will be plot A, that in the bottom left will be plot B, that in the top right will be plot C, that in the center right will be plot D, and that in the bottom right will be plot E.

The input signal is composed of 3 different sub-signals x_1, x_2, and x_3, each of which has a certain frequency band. It is important to know which sub-signal corresponds to which frequency "bubble" in plot A. We can do this by comparing the apparent frequencies of each of these bubbles to the frequencies presented in each equation. Here, the leftmost bubble corresponds to the highest-frequency equation: x_3. The middle bubble corresponds to the lowest-frequency equation: x_1. Finally, the rightmost bubble corresponds to equation x_2.

Plot B then shows us the frequencies present in the overall signal. The peaks at $\pm 0.8\pi$ correspond to x_3, for example. Plot C, in turn, presents us with the way in which the magnitude of the input signal is amplified given the component frequencies. It is the **magnitude response**. Plot D, similarly, presents us with the way in which the phase of the input signal is shifted given the component frequencies. It is the **group delay**, which was found by taking the negative derivative of the **phase response** (not shown in any of the diagrams).

The output signal, shown in plot E, can then be constructed using the information provided in plots A, C and D as follows:

For $\omega = \pm 0.2\pi$:

- Magnitude response is approximately 1.2 → x_1 is amplified by a factor of 1.2
- Group delay is 150 → x_1 is shifted 150 time units to the right

For $\omega = \pm 0.4\pi$:

- Magnitude response is 2 → x_2 is amplified by a factor of 2
- Group delay is 10 → x_2 is shifted 10 time units to the right

For $\omega = \pm 0.8\pi$:

- Magnitude response is close to zero → x_3 is almost fully attenuated
- Group delay is -50 → x_3 is shifted 50 time units to the left

Bode Diagrams Defined

From previous chapters, you might have noticed how difficult it can be to plot magnitude and phase responses on to-scale drawings. Bode Diagrams are a method for plotting and visualizing magnitude and phase responses using logarithmic scales for the frequency range and the magnitude response value.

A bode diagram of a system consists two plots:

1) Bode Magnitude Plot: $20\log_{10}\left(|H(j\omega)|\right)$ along a logarithmic scale of ω
2) Bode Phase Plot: Simply the phase response on a logarithmic scale of ω

Advantages:

1) A larger range of frequencies can be analyzed.
2) The logarithm property allows us to **sum** different subcomponents of a complicated frequency response magnitude instead of multiplying them, thus simplifying calculations.

First-Order Terms

A first-order term is a term that has one of the following formats:

$1+\frac{j\omega}{\omega_0}$ (where ω_0 is a constant term known as the *natural frequency*)

$1-\frac{j\omega}{\omega_0}$

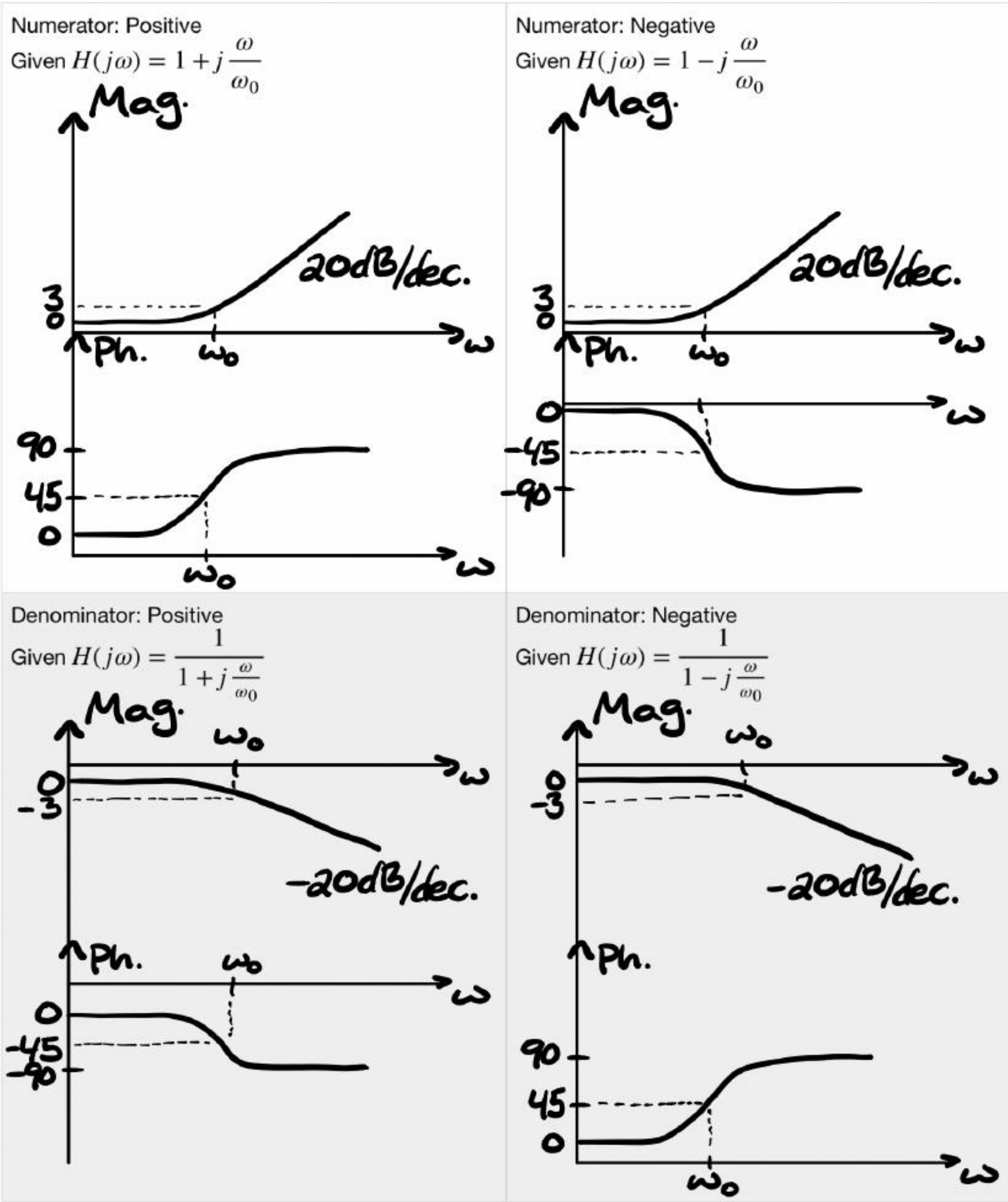
$\frac{1}{1+\frac{j\omega}{\omega_0}}$

$\frac{1}{1-\frac{j\omega}{\omega_0}}$

When plotting the Bode Diagrams for each of these cases, we must consider three different regions along the frequency axis and calculate the Bode magnitude and phase in each of the regions. From those calculations, the Bode Plots can be constructed. Feel free to fill in the following table when constructing your own Bode Diagram.

Region	Bode Magnitude: $20\ \log_{10}\left(|H(j\omega)|\right)$	Bode Phase: $\angle H(j\omega)$ (same as usual phase)
$\omega = 0$		
$\omega = \omega_0$		
$\omega >> \omega_0$		

Example plots for each of the four possible basic options listed above are shown below. Remember that all frequency axes are scaled logarithmically.



Second-Order Terms

In the context of second-order terms, we introduce the damping coefficient, ζ. This term appears when the system could be defined by complex poles. There are four possible cases when dealing with the damping coefficient:

- When $\zeta = 1$, there is critical damping.
- When $0 < \zeta < 1$, there is underdamping and oscillations appear.
- When $\zeta > 1$, there is overdamping and there are real poles.
- When $\zeta < 0$, the system is unstable.

The general form of the second-order term is one of the following:

$$H(j\omega) = \left(\frac{j\omega}{\omega_0}\right)^2 + 2\zeta\left(\frac{j\omega}{\omega_0}\right) + 1$$ (numerator form)

$$H(j\omega) = \frac{1}{\left(\frac{j\omega}{\omega_0}\right)^2 + 2\zeta\left(\frac{j\omega}{\omega_0}\right) + 1} = \frac{\omega_0^2}{(j\omega)^2 + 2\zeta\omega_0 j\omega + \omega_0^2}$$ (denominator form)

When the system is underdamped, oscillations take place. Additionally, when $0 < \zeta < \frac{1}{\sqrt{2}}$ a resonant peak forms at the resonant frequency:
$\omega_r = \omega_0\sqrt{1 - 2\zeta^2}$

At the resonant frequency, we observe either a small dip or peak in the Bode Magnitude Plot, where:

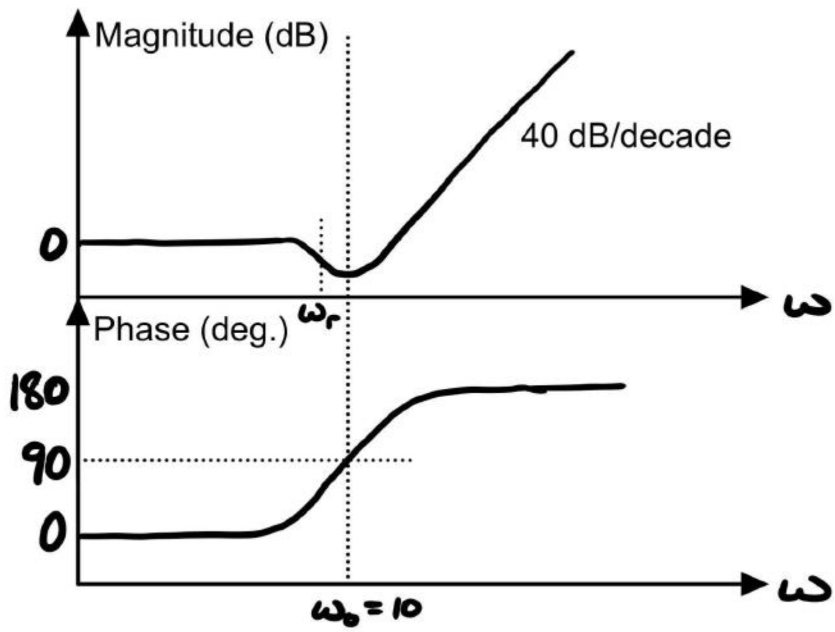
$$\text{Magnitude} = +20\log_{10}\left(2\zeta\sqrt{1-2\zeta^2}\right)$$ (numerator form)

$$\text{Magnitude} = -20\log_{10}\left(2\zeta\sqrt{1-2\zeta^2}\right)$$ (denominator form)

When plotting the Bode Diagrams for a second-order term, we must check the 3 regions that are checked usually in First Order Terms in addition to the resonant frequency ($\omega = \omega_r$).

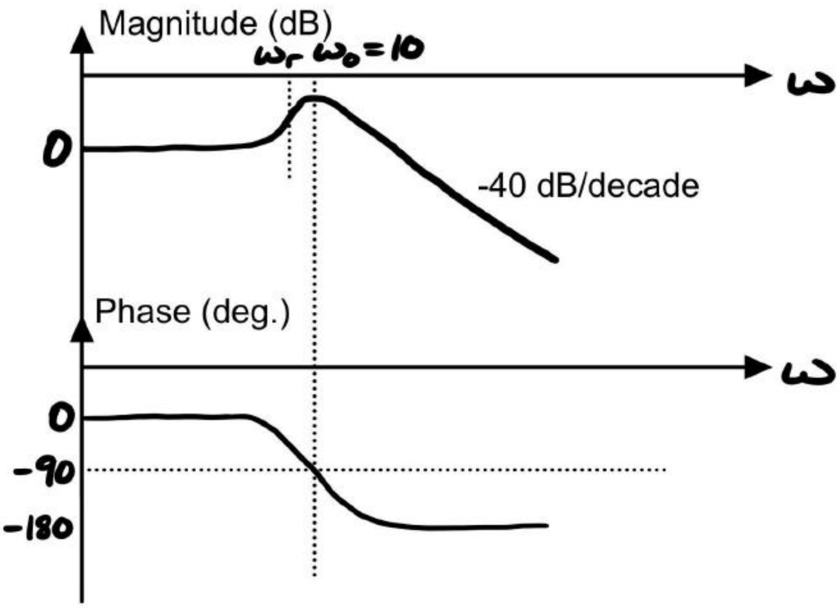
For example, given a “numerator” form (shown below), we can draw a corresponding Bode diagram:

$$H(j\omega) = \left(\frac{j\omega}{10}\right)^2 + 0.2\left(\frac{j\omega}{10}\right) + 1$$



Given a “denominator” form (shown below), we can draw a corresponding Bode diagram:

$$H(j\omega) = \frac{1}{\left(\frac{j\omega}{10}\right)^2 + 0.2\left(\frac{j\omega}{10}\right) + 1}$$



A Word on Constants

Drawing Bode diagrams for contents is fairly easy, yet frequently neglected by students on exams.

The Bode Magnitude for a constant k is a horizontal line at $-20\log_{10} k$.
The Bode Phase is always zero for the constant term.

General Approach to Solving a Bode Diagram Problem

The most common problems in this chapter can either:

A) Provide you with a complicated frequency response $H(j\omega)$ and ask for the Bode Diagrams
B) Provide you with Bode Diagrams and ask you to match the appropriate Bode Plot from a given set of options (Multiple Choice Question)

For the first type of problems:

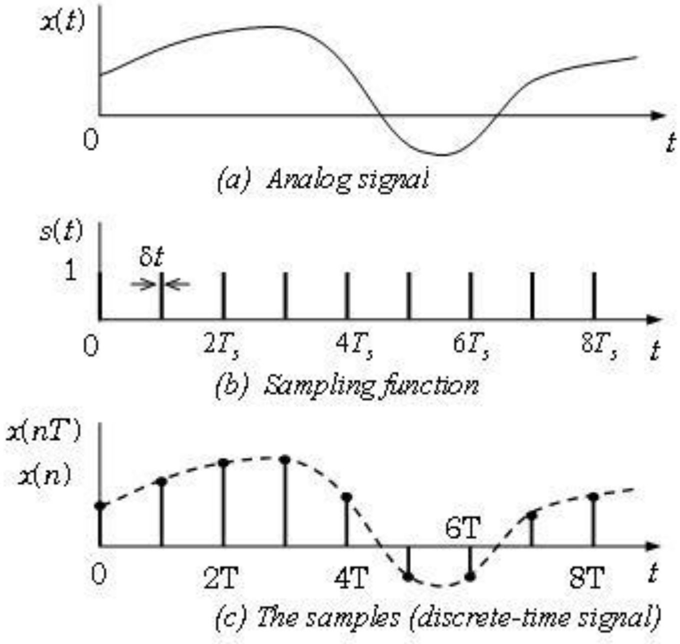
1) Break down the frequency response into simpler terms: $H(j\omega) = H_1(j\omega)H_2(j\omega)\ldots$. Polynomial Factorization and Partial-Fraction Decomposition are common techniques used to complete this simplification step.
2) Classify each term as either 1st order, 2nd order, or a constant and draw the Bode Plots for each term using the rules for their respective category.
3) Add the plots for all individual terms (a good approach is to start from $\omega = 0$ and move in increasing frequency).

There is no general approach to solving the second type of problems, as they require more intuition from the student. However, it is important to keep track of the following:

1) The natural frequencies of each of the terms in the different options for the frequency responses.
2) The initial value of the Bode Plots when $\omega = 0$ (usually easy to calculate for each option).
3) The final trend in the Bode Plots when the frequency becomes larger than all natural frequencies (this should be easy to determine for each option).
4) The cancelling effect that can take place between 2 terms (e.g. if a term has a slope of +20 and the other term has a slope of -20 and their natural frequencies are close to each other).

Introduction to Sampling and Reconstruction

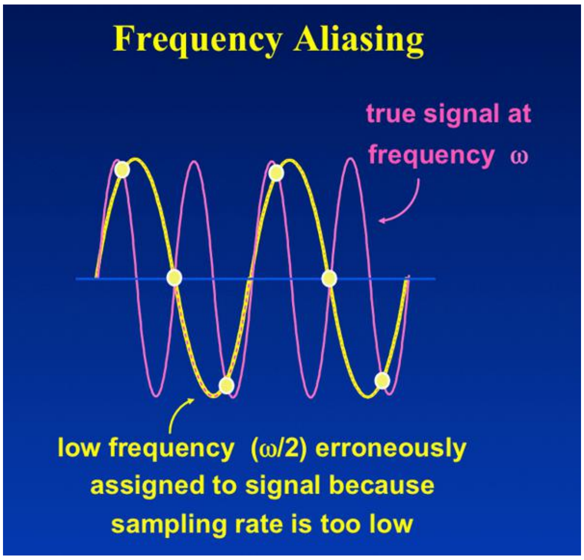
Sampling is the process of transforming a CT signal to a DT signal. This is done by selecting certain points from the CT signal at a constant sampling rate Ω_s.We can think of this operation as a convolution between the CT signal and a train of impulses known as the sampling function.



https://signalprocessingsampling.weebly.com/

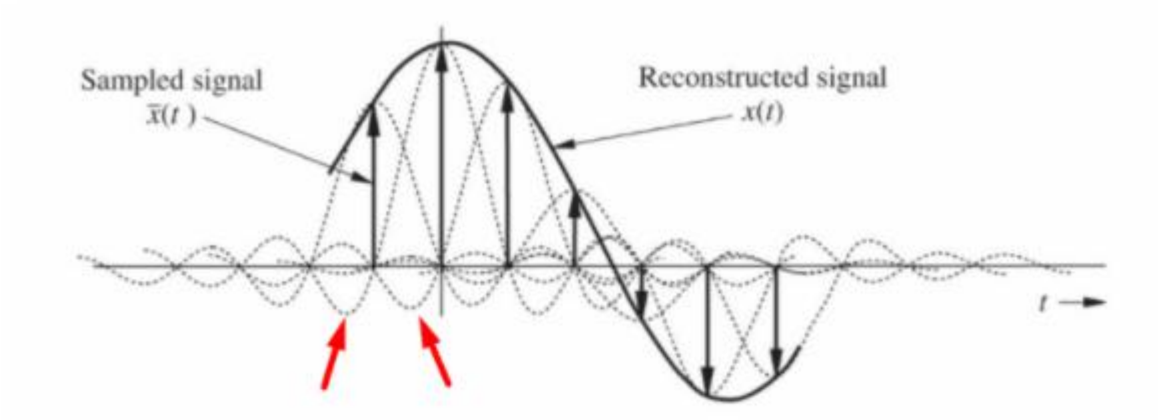
Aliasing is a phenomenon that occurs when we sample at a very slow rate and start losing some of the frequencies found in the original CT signal.

Thus, when sampling, one must take into account the frequency domain of the CT signal and in particular, the highest frequency that exists in this signal, also known as the Nyquist Frequency Ω_N. To avoid aliasing, we must choose a sampling frequency so that $\Omega_s \geq 2\Omega_N$



http://mriquestions.com/uploads/3/4/5/7/34572113/6443188_orig.gif

Reconstruction is the process of transforming a DT signal into a CT signal. This is done through interpolating the discrete points and joining them in an intelligent way to create a CT signal. The 'joining' process is usually done through passing a filter that helps us fill in the gaps between the points of the DT signal. Using a filter can also allow us to remove any 'unwanted' frequencies at the edge by adding a cutoff. Reconstruction can also have its own frequency Ω_r.No aliasing happens during reconstruction.



https://www.hebergementwebs.com/analog-communication-tutorial/analog-communication-sampling

Why Sampling?

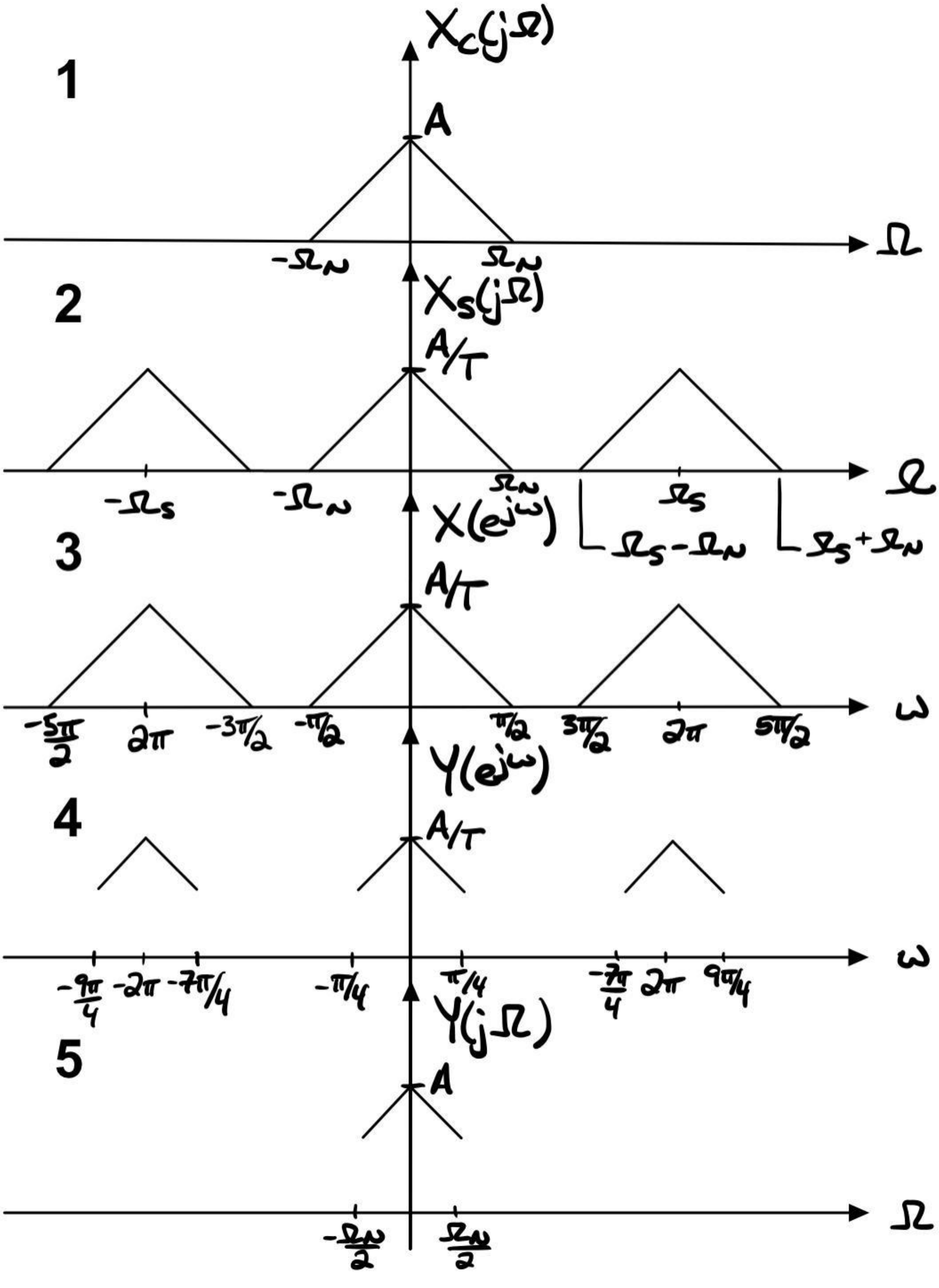
Computers and digital electronics handle data in discrete bits. It is also impossible for a computer to capture and store a CT signal from nature, as it would require infinite memory for the data (e.g. the computer will need to know the signal at t=0.1s, 0.01s, 0.001s….). Hence, we need to limit the time frequency at which the data is stores, and this is done through sampling (e.g. using $\Omega_s = 10Hz$, we would only store the points at t=0.1s,t=0.2s, t=0.3s ….).

Note on the Notations for this Chapter

This chapter will deal with frequencies both for DT signals and CT signals. To avoid confusion, we use Ω for the frequency domain of CT signals and ω for the frequency domain of DT signals.
To move between Ω and ω, we use the simple relation $\omega = T\Omega$, where T is either the sampling or reconstruction period $\left(\frac{2\pi}{\Omega_s}\ or\ \frac{2\pi}{\Omega_r}\right)$ depending on whether we are sampling or reconstructing.

Sampling and Reconstruction in the Frequency Domain Explained Through an Example

Given the spectrum $X_C(j\omega)$, $\Omega_S = 4\Omega_N$, and A, the amplitude at $\Omega = 0$ for the input spectrum:



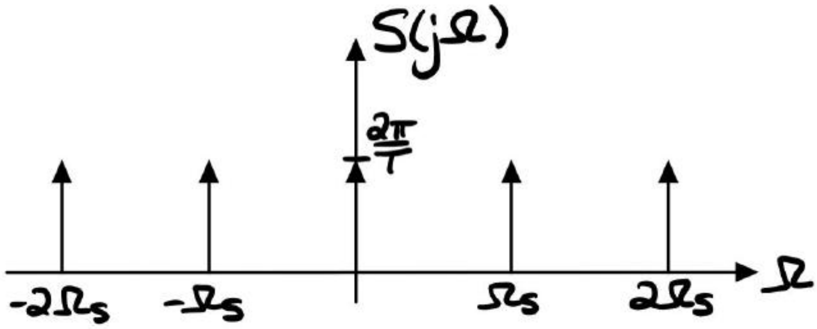
The plots shown above can be explained as follows:

Plot 1

This is the input spectrum, the range of frequencies being introduced to the system.

Plot 2

This is the sampled signal's spectrum. It is the result of convolving $X_C(j\Omega)$ and the impulse train sampling function, $S(j\Omega)$:

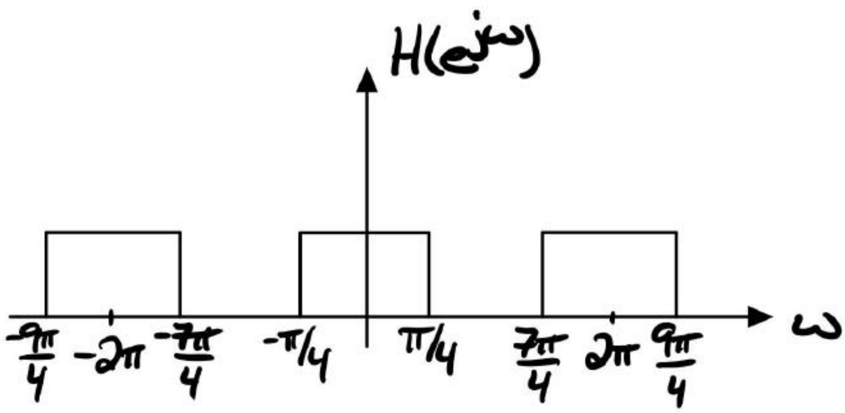


Plot 3

This is the discretized version of the sampled signal's spectrum. Ω_S is **always** mapped to 2π, and Ω_N is mapped according to $\Omega_S = 4\Omega_N$. Here, this mapping corresponds to $\Omega_N = \frac{\pi}{2}$.

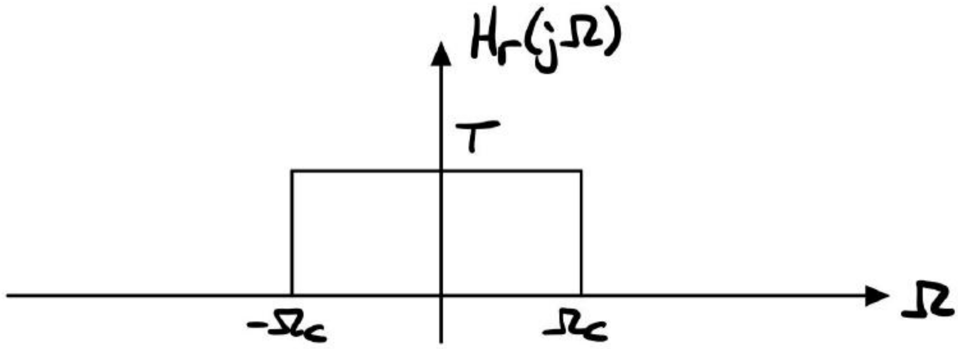
Plot 4

This is the discretized sample signal's spectrum after passing through an arbitrary discrete filter, $H\left(e^{j\omega}\right)$ (shown below). Note that this is an optional step.



Plot 5

This is the reconstructed output spectrum that results from multiplying $Y\left(e^{j\omega}\right)$ with the following continuous lowpass reconstruction filter, $H_r(j\Omega)$. Note that, here, $\Omega_N < \Omega_C < (\Omega_S - \Omega_N)$.



Impulse Invariance Method

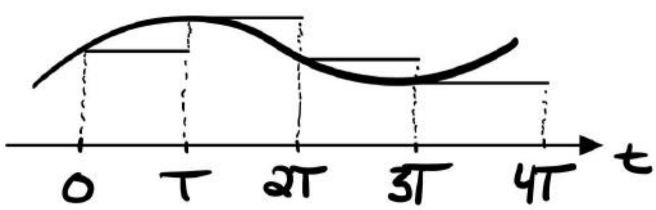
Impulse invariance is a technique used to turn CT systems of **finite frequency bandwidth** into DT systems. It follows the steps below:

1) Find the impulse response $h_c(t)$ of the CT system (sometimes given)
2) Use the following relation to find the impulse response of the DT system: $h_d[n] = Th_c(nT)$ where T is the sampling period.
3) Using a fourier transform, or any other technique, we can find the frequency response $H_d\left(e^{j\omega}\right)$ of the DT system.

Note: just as in sampling for signals, we must choose a sampling rate that is large enough so that all frequencies in the CT system are captured (i.e. to avoid aliasing).

Practical Sampling

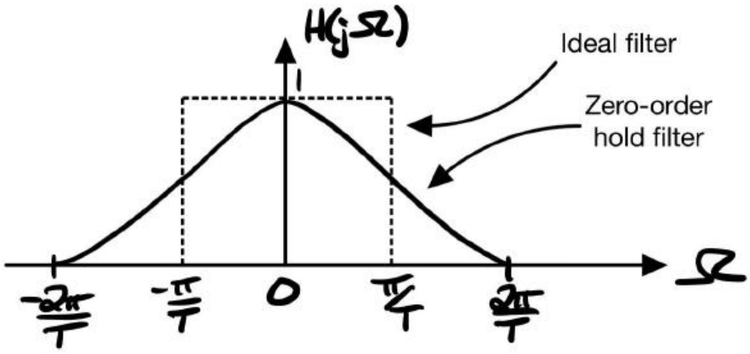
Practical sampling is a realistic version of sampling and reconstruction that allows us to tune the accuracy of the sampling. Our aim here is to capture a CT signal in nature and represent it in the computer's environment. At its coarsest level, we find the zero-order hold: the input signal is converted to a series of "steps".



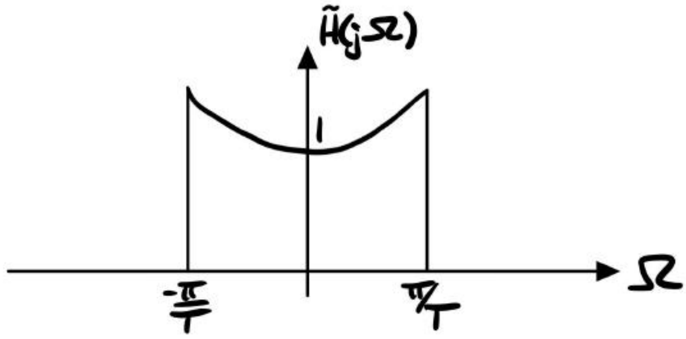
This is done using a periodic sampling function defined as:

$$h(t) = \begin{cases} 1 & \text{over T} \\ 0 & \text{otherwise} \end{cases}$$

In the frequency domain, this sampling function takes on the following shape:



To eliminate the error created by the sampling function, reconstruction is done using the following reconstruction filter. Notice how the filter keeps the peak at 1, boosts the frequencies that are lower than those in the ideal filter, and completely cuts off the frequencies beyond those permitted by the ideal filter.



We can then expand this concept to the first-order hold, which involves linear interpolation (instead of “steps”), or even the third-order hold, which involves cubic spline interpolation.

Resampling

Resampling is a process that modifies the sampling rate of an already-sampled signal in Discrete Time.

Downsampling involves the following procedures in the time-domain:

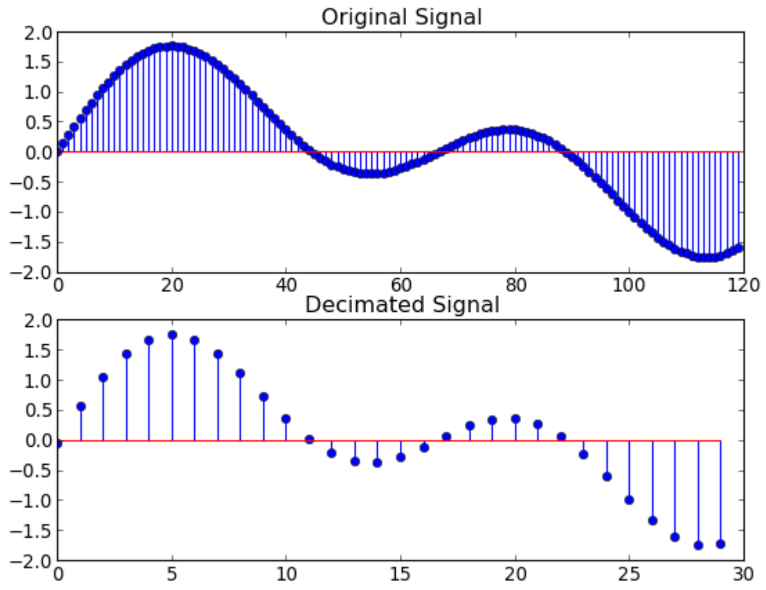
1) ‘Decimating’ data points of a DT signal at a rate of M
2) Rearranging the data points so that they fill in the gaps. For example, if M is 10, then for every 10 consecutive data points, only one is kept and then the surviving datapoint at t=10s is moved to t=1s, the datapoint at t=20s is moved to t=2s and so on.

In the frequency domain, this results in the following:

1) Stretching the frequency spectrum by a factor of M
2) Downscaling the original amplitude of each frequency by a factor of M (i.e. multiply by 1/M)

If the downsampling rate M is too high, this might also result in aliasing since decimating too many points would lead to losing certain high frequencies from the original signal. Hence, when sampling is done followed by downsampling, the following relation must be satisfied to avoid aliasing: $\Omega_s \geq 2M\Omega_N$

Example of Downsampling with M=4



http://mubeta06.github.io/python/sp/multirate.html

.

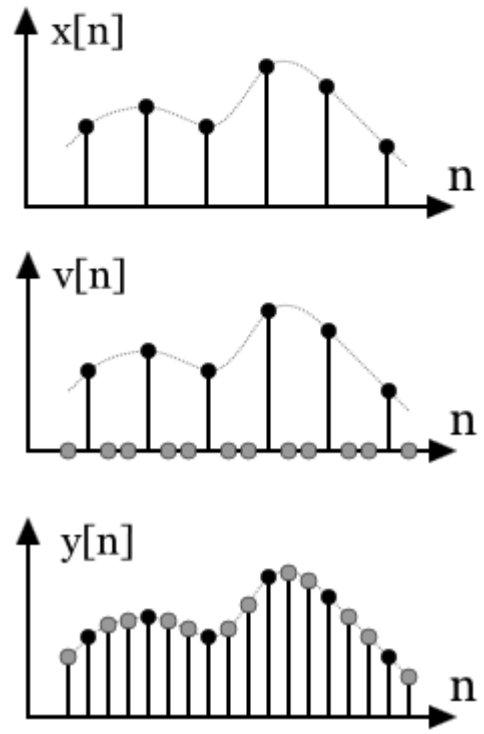
Upsampling is the opposite of downsampling. It involves the following steps in the time domain:

1) Space out the existing data points L spaces apart (social distancing).
2) Interpolate between the original data points to fill in the data for the newly created points.

In the frequency domain, this results in the following:

1) Compressing the frequency spectrum by a factor of L
2) Multiplying the amplitude of each frequency by a factor of L.

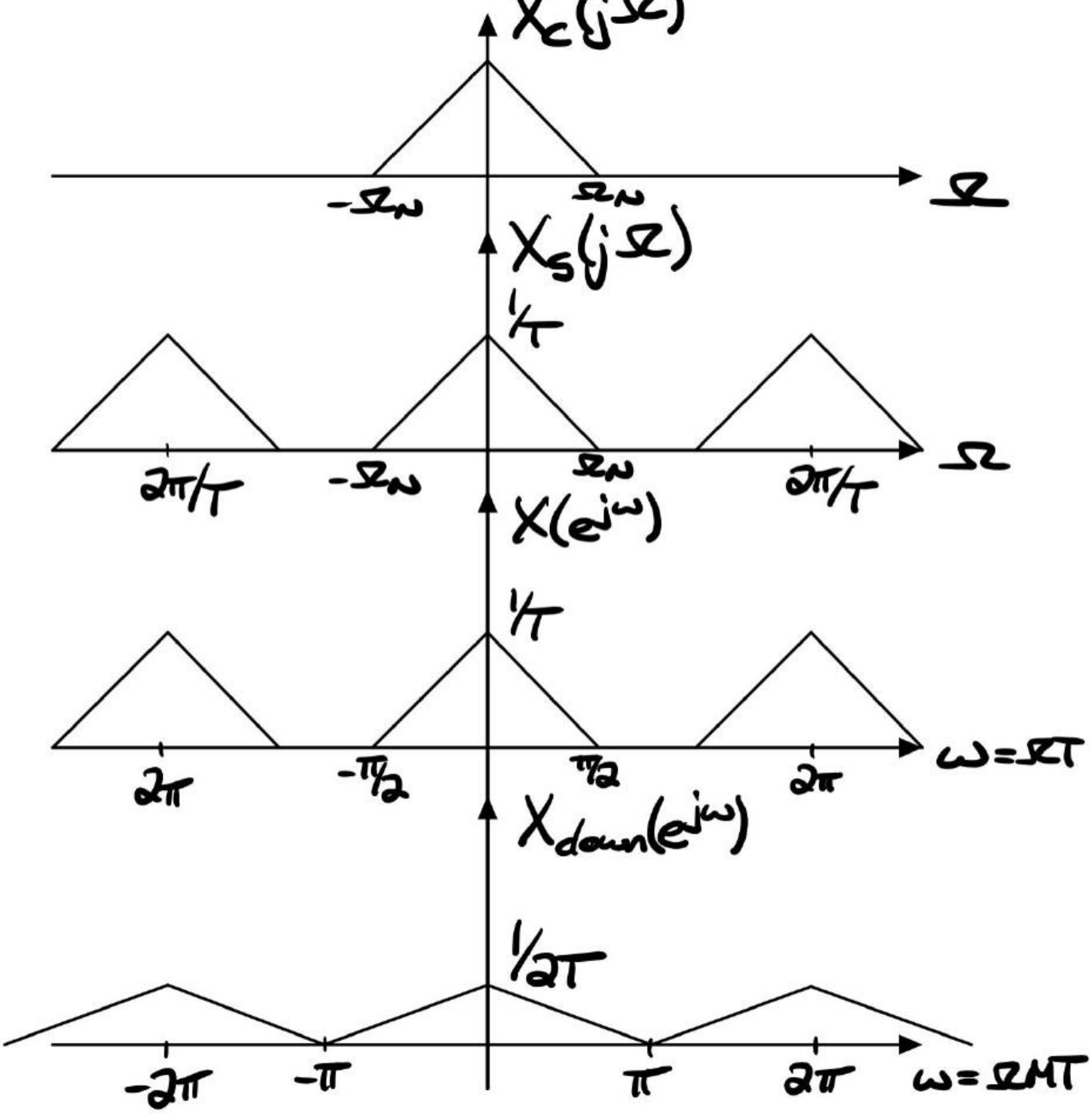
Aliasing never happens during upsampling.



https://www.divilabs.com/2014/07/upsampling-interpolation-of-discrete.html

Resampling in the Frequency Domain Explained by Examples

Using $M = 2$,



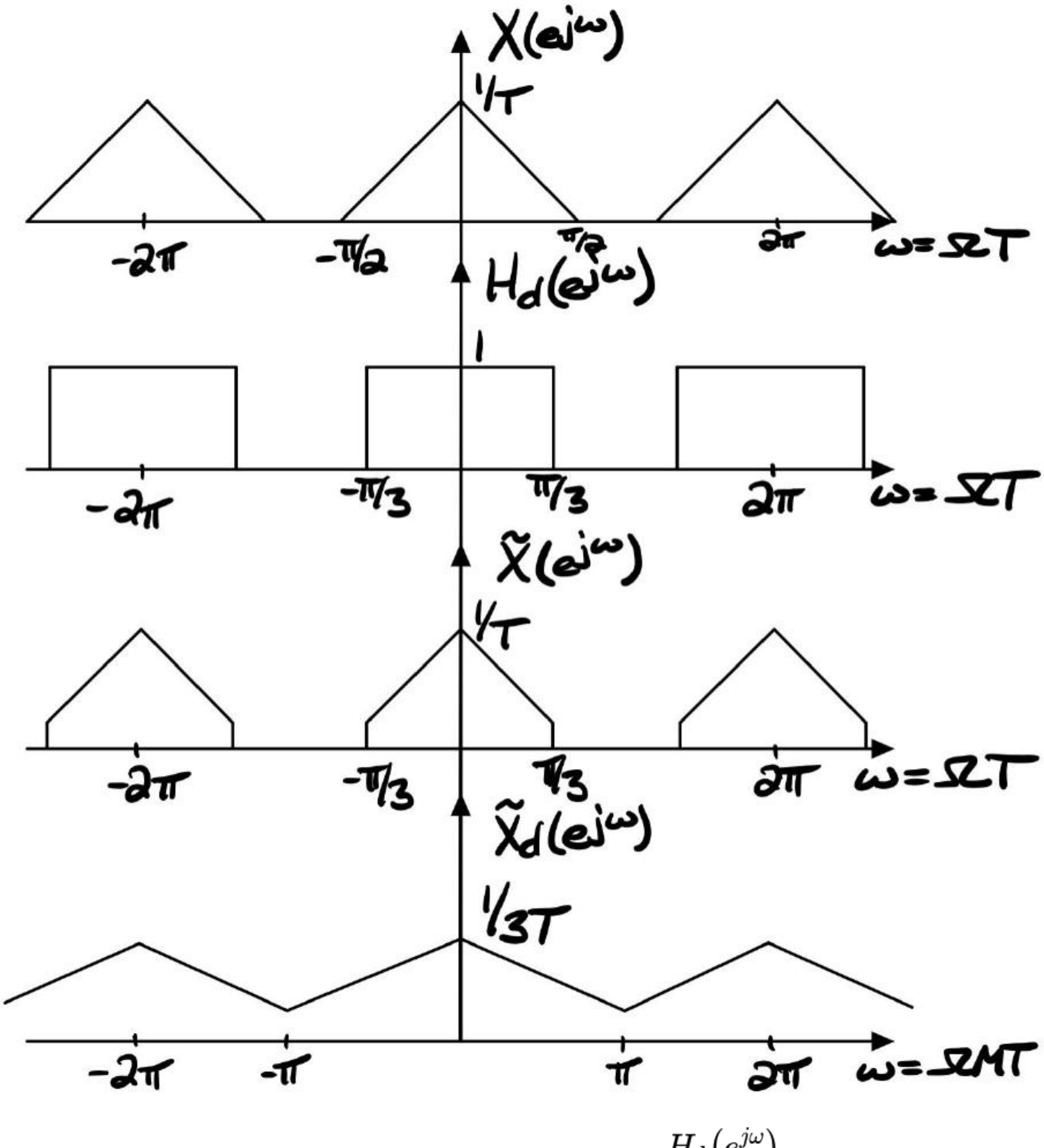
Notice how the amplitude of the downsampled discretized signal, $X_{\text{down}}\left(e^{j\omega}\right)$, is scaled by a multiple of $\frac{1}{M} = \frac{1}{2}$.

What would have happened had there been aliasing? Recall that, to avoid aliasing, we need $\Omega_S \geq 2M\Omega_N$. Consider the case where $M = 3$ and $\Omega_S = 4\Omega_N$:

$\Rightarrow \Omega_S = 2(3)\Omega_N$

$\Rightarrow \Omega_S = 4\Omega_N \ngeq 6\Omega_N$

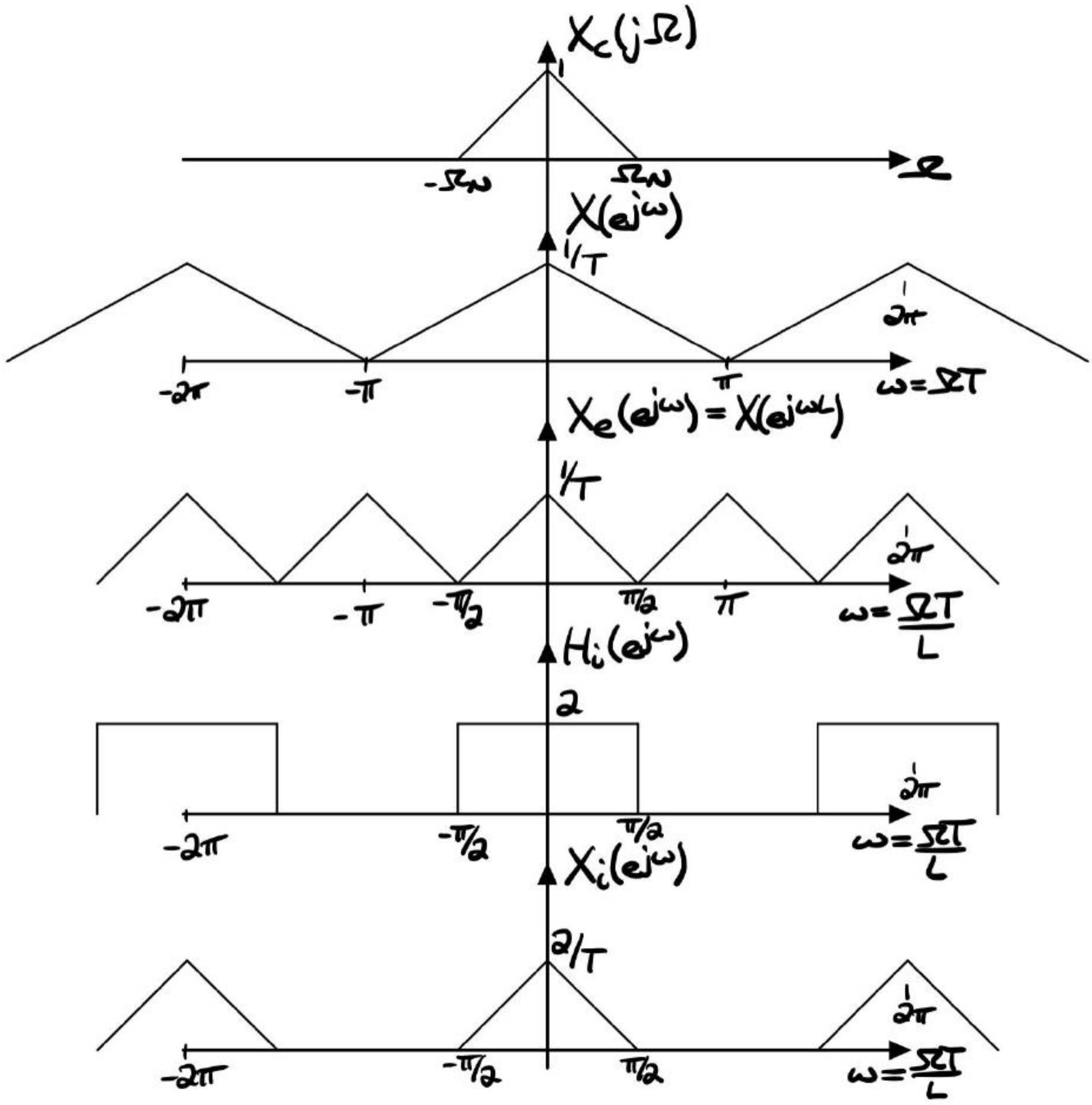
We will have aliasing in this case. We can use an antialiasing lowpass filter to remove aliasing at the expense of signal information by using $\omega_C = \frac{\pi}{M}$. This is called decimation.



Notice how using the appropriate discrete cutoff filter, $H_\mathrm{d}\left(e^{j\omega}\right)$, we force the input spectrum to obey the $\Omega_S \geq 2M\Omega_N$ rule. Then, when downsampling the modified input spectrum, $\tilde{X}\left(e^{j\omega}\right)$, we see that there is no aliasing. However, information is lost relative to the original spectrum. You may be wondering: we lost information either way, so, what's the point? Remember that aliasing is specifically the loss of information due to downsampling. By using a cutoff filter, we are removing certain information intentionally to avoid overlaps, which may lead to a distorted signal.

As for upsampling, as mentioned previously, we never need to worry about aliasing because we are adding new samples. As such, information cannot be lost due to overlaps (as we are “adding space”, not “removing space”).

Consider the case where $L = 2$ and $\Omega_S = 2\Omega_N$:



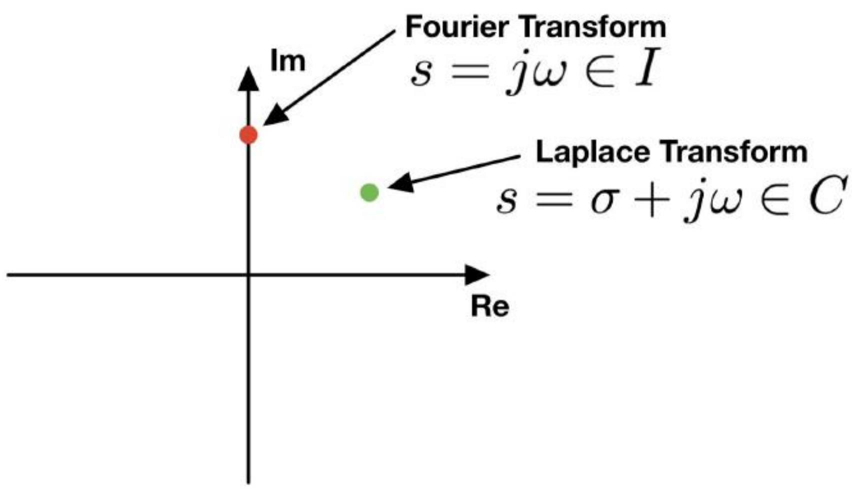
Notice how the amplitude of the signal has been multiplied by a factor of $L = 2$. The upsampled input spectrum, $X_{\mathrm{i}}\left(e^{j\omega}\right)$, now features more “space” for new samples to be taken. In fact, the input spectrum has been sampled *as if* we had used $\Omega_S = 2L\Omega_N = 4\Omega_N$.

Introduction to the Laplace Transform

The Laplace Transform is a type of transform that maps a CT signal or impulse response of a CT system from the time domain into a complex number space 's' defined as $s = \sigma + j\omega$. The equation that defines this transform is the following:

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

Similar to the Fourier Transform, there is a table of Laplace Transform Pairs that makes the conversion easier than solving the integral. In fact, the Laplace Transform is a generalization of the Fourier Transform.



Why Laplace and not Fourier?

1) Laplace transform can be applied on signals and systems that otherwise do not have an existing Fourier Transform (aperiodic non-dirichlet signals, unstable LTI systems, LTI systems with non-zero initial conditions)
2) The Laplace transform can give us more information about the signal thanks to the 2 dimensional real-imaginary space it has $(\sigma \text{ and } \omega \text{ axes})$. It can tell us about both exponential and oscillatory behavior in the time domain. On the other hand, the Fourier Transform only stores information about the oscillatory components of the signal/system in the time domain. In other words, it only has a 1 dimensional space composed of the ω axis.

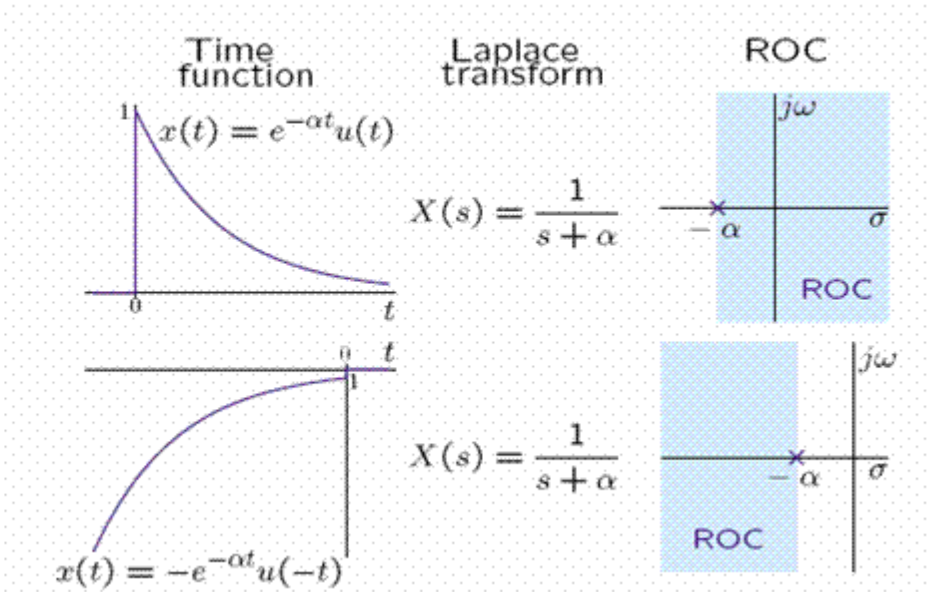
The Region of Convergence

Since the formula for the Laplace transform is an improper integral, it may diverge in some cases. Hence, only certain values of s will make the integral converge in those cases. The range of values of s that make the integral converge are known as the REGION OF CONVERGENCE (ROC). The region of convergence is a rectangular region that sweeps along the real axis of the real-imaginary plane.

Usually, the Laplace Pair Tables provide the region of convergence for each transform. We can also tell the region of convergence of a simple signal (1 pole) using the following methodology:

1) Find the value of s that 'blows up' the Laplace Transform expression. This value is known as the pole.
2) If the signal is right-handed, the ROC is the half-plane that lies to the right of that pole. Otherwise if the signal is left-handed, then the ROC is the half-plane that lies to the left of the pole.

Below is an example for the ROC of 2 simple signals that have the same Laplace Transform Expression, one right-handed and the other left-handed.



For a signal that has more than 1 pole, the following steps are used to find its ROC:

1) Separate the signal into simpler signals, each having 1 pole.
2) FInd each of those poles.
3) Find the ROC of each simple signal from its corresponding pole.
4) The ROC of the complex signal is the intersection of the ROC's of the simpler signals.

The ROC for H(s) of an impulse response h(t) can also reveal some of the properties of a system:

A) If the ROC is right handed (goes to $\sigma = \infty$), then the system is CAUSAL.
B) If the ROC includes the imaginary axis, then the system is BIBO STABLE.

Poles and Zeros

The Laplace Transform of any given signal/system can be written in the following form:

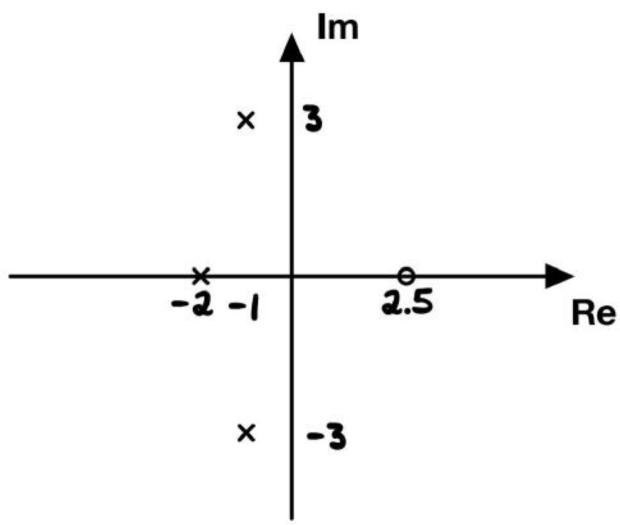
$$X(s) = \frac{numerator}{denominator}$$

The values of s that cause the numerator to go to zero are known as the ZEROS.

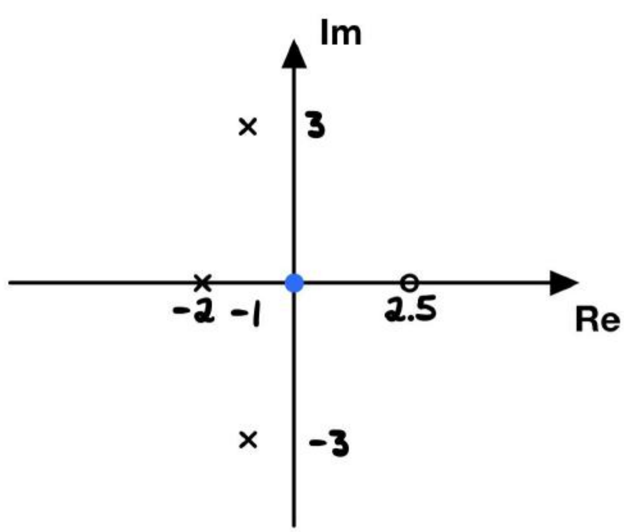
The values of s that cause the denominator to go to zero (and hence X(s) to blow up) are the POLES.
We use pole-zero maps to represent those points by simply plotting them in their respective positions on the real-imaginary plane.
As mentioned earlier, poles play a key role in determining the region of convergence.
Note that in the case where a pole and zero are located in the exact same spot they cancel each other.

Sketching Magnitude Response from Pole-Zero Map

The magnitude response can be quickly sketched from the pole-zero map. Take the following map, for example. Recall that poles are indicated by X's, while zeros are indicated by O's.



Start at $\omega = 0$.

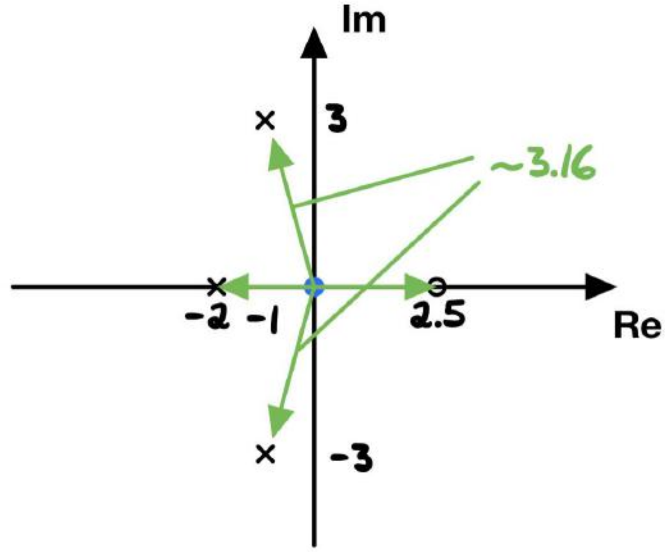


Look at the distances from this center point to every pole and every zero. Calculate $|H(s)|$ with the following equation:

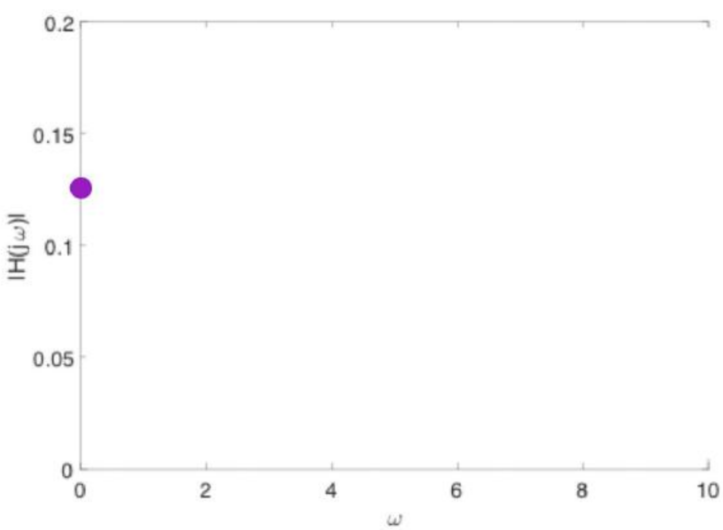
$$|H(s)| = \frac{\text{product of distances to the zeroes}}{\text{product of distances to the poles}}$$

Here, we get:

$$|H(s)| = \frac{2.5}{2 \cdot 3.16 \cdot 3.16} \approx 0.125$$



This gives us a starting point for our magnitude response sketch.

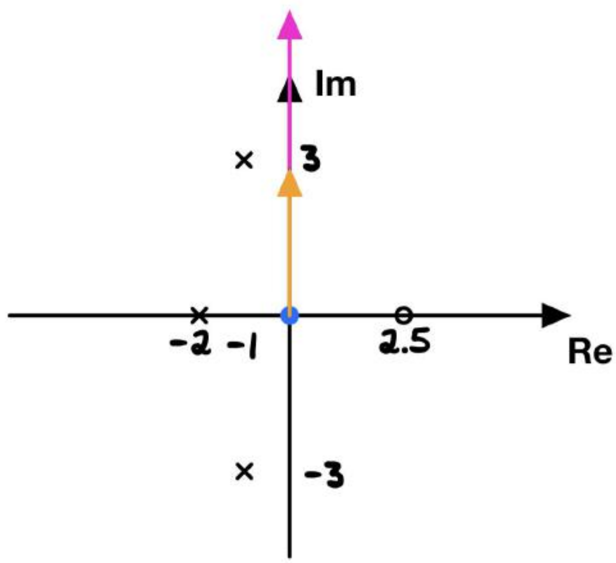


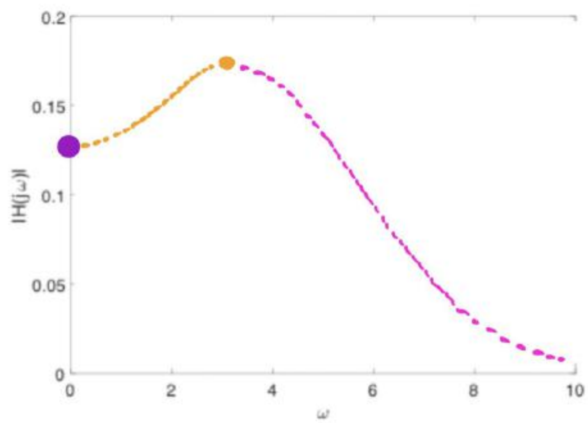
Next, move upwards along the imaginary axis. Roughly keep track of the distances from this point along the imaginary axis to every pole and every zero.

First, consider the orange segment. We get closer to a pole and get further from a zero. As a result, the magnitude increases.

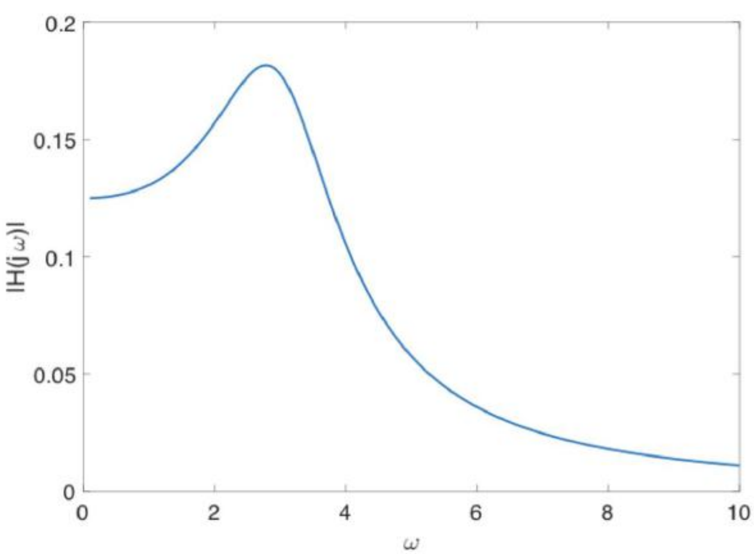
$$|H(s)| = \frac{\text{product of distances to the zeroes} \Uparrow}{\text{product of distances to the poles} \Downarrow} \Uparrow$$

Then, consider the pink segment. We get perpetually further from all poles and zeros, leading to a decay down to zero.





Notice that this matches the MATLAB-generated plot!



Inverse Laplace Transform

The Inverse Laplace Transform has the following complicated form:

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

Note that in BIEN350, we are not required to solve this type of integral. Hence, to do inverse Laplace, we always use the Laplace Transform Pair Table.

Properties of the Laplace Transform

PROPERTY	TIME DOMAIN	S-DOMAIN	REGION OF CONVERGENCE
Linearity	$\alpha x(t) + \beta y(t)$	$\alpha X(s) + \beta Y(s)$	At least the intersection of the ROCs of X(s), Y(s). The ROC may **not exist**, may be an **intersection** and/or may be **extended** by pole-zero cancellation.
Time Shifting	1. $x(t-t_0)$ — Time domain 2. $e^{s_0 t}x(t)$ — S-domain	1. $e^{-st_0}X(s)$ 2. $X(s-s_0)$	Unchanged.
Time Scaling	$x(\alpha t)$	$\frac{1}{\lvert\alpha\rvert}X(\frac{s}{\alpha})$	ROC: αR, with R, the original region of convergence. - $\lvert\alpha\rvert < 1$: Compression - $\lvert\alpha\rvert > 1$: Extension - $\alpha < 0$: Flipping, with compression or extension depending on the above two statements.
Time Inversion	$x(-t)$	$X(-s)$	See time scaling property, $\alpha = -1$.
Conjugate Symmetry	$x^*(t)$	$X^*(s^*)$ —> Poles and zeroes come in conjugate pairs.	Unchanged.
Convolution	$x_1(t) \circledast x_2(t)$	$X_1(s)\cdot X_2(s)$	See linearity property.
Differentiation	1. $\frac{dx(t)}{dt}$ — Time domain 2. $-tx(t)$ — S-domain	1. $sX(s)$ 2. $\frac{dX(s)}{ds}$	1. ROC of the derivative is **extended** if it has a pole at zero. 2. Unchanged.
Integration	$\int_{-\infty}^{t} x(\tau)d\tau$	$\frac{X(s)}{s}$	ROC: intersection of the original ROC with the right half plane, $\mathbb{R}\{s\} > 0$
Initial and Final Value Theorems	Given $x(t) = 0, t < 0$, where $x(t)$ does not have an impulse function at $t = 0$.	1. $x(0^+) = \lim_{s\to\infty} sX(s)$ (Initial Value) 2. $\lim_{t\to\infty} x(t) = \lim_{s\to 0} sX(s)$ (Final Value)	N/A

Unilateral Laplace Transform

The Unilateral Laplace Transform is a variant of the Laplace Transform defined as the following:

$$X(s) = \int_{0-}^{\infty} x(t)\, e^{-st}\, dt$$

The ROC of such a transform is always a right-half plane.

The Unilateral Laplace Transform is used ubiquitously in solving ODE's due to the following properties that it gives for derivative terms:

$$\frac{dx(t)}{dt} \to sX(s) - x\left(0^-\right)$$

$$\frac{d^2x(t)}{dt^2} \to s^2X(s) - sx\left(0^-\right) - \frac{dx(0^-)}{dt} - \frac{d^{n-1}x(0^-)}{dt^{n-1}}$$

$$\frac{d^nx(t)}{dt^n} \to s^nX(s) - s^{n-1}x\left(0^-\right) - s^{n-2}x'\left(0^-\right) \ldots\ldots -s\,\frac{d^{n-2}x(0^-)}{dt^{n-2}} - \frac{d^{n-1}x(0^-)}{dt^{n-1}}$$

Solving an ODE Using The Unilateral Laplace Transform

Example:

$$y''(t) + 3y'(t) + 2y(t) = x(t)$$

$$x(t) = 2u(t)\,,\; y\left(0^-\right) = 3,\; y'\left(0^-\right) = -5$$

1) Take the unilateral Laplace Transform of both sides of the equation. Pro tip: keep all the initial condition and input function constants as variables until the very end.

$$s^2Y(s) - sy\left(0^-\right) - y\left(0^-\right) + 3sY(s) - 3y\left(0^-\right) + 2Y(s) = X(s)$$

2) Isolate for the term Y(s)

$$Y(s) = \frac{y(0^-)(s+3)}{(s+1)(s+2)} + \frac{y'(0^-)}{(s+1)(s+2)} + \frac{X(s)}{(s+1)(s+2)}$$

3) Break down the other side of the equation into Zero State and Zero Input Responses (write it as a sum of both). A good way to do this is to set the term that came from X(s) to zero and the surviving terms will make up the zero-input response. The terms that were not included in the zero-input response are the zero-state response. Another approach is to set the initial conditions to zero and collect the surviving terms to find the zero-state response.

$$Y_{ZI}(s) = \frac{y(0^-)(s+3)}{(s+1)(s+2)} + \frac{y'(0^-)}{(s+1)(s+2)} = \frac{3(s+3)-5}{(s+1)(s+2)}$$

$$Y_{ZS}(s) = \frac{X(s)}{(s+1)(s+2)} = \frac{2}{s(s+1)(s+2)}$$

4) Take the Inverse Laplace of both sides of the equation. Y(s) becomes y(t) and the other side of the equation is the answer as a sum of zero-input and zero-state responses $y_{zi}(t)$ and $y_{zs}(t)$.

$$Y_{ZI}(s) = \frac{1}{(s+1)} + \frac{2}{(s+2)}$$ (after fraction decomposition)

$$y_{zi}(t) = u(t)\left[e^{-t} + 2e^{-t}\right]$$

$$Y_{ZS}(s) = \frac{1}{s} - \frac{2}{(s+1)} + \frac{1}{(s+2)}$$

$$y_{zs}(t) = u(t)\left[1 - 2e^{-t} + e^{-2t}\right]$$

5) Combine both zero-state and zero-input solutions to get the final solution:

$$y(t) = y_{zi}(t) + y_{zs}(t) = u(t)\left[1 - 1e^{-t} + 3e^{-2t}\right]$$

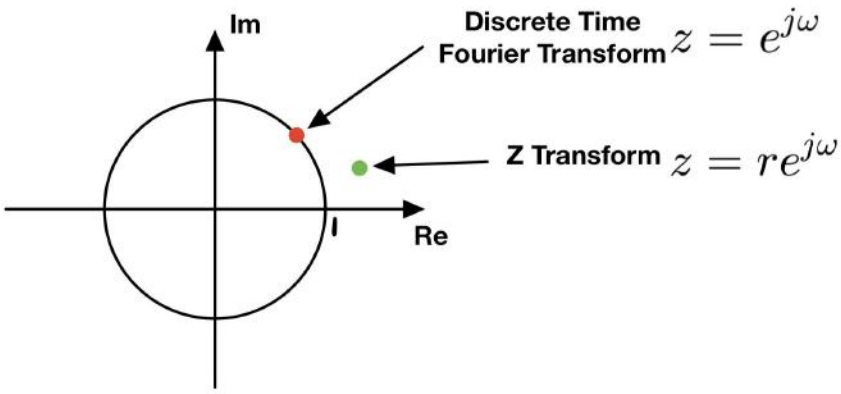
Z-Transform Introduction

The Z-Transform is a type of transform that maps a DT signal or impulse response of a system from the time domain into a complex number space consisting of the real and imaginary axes. The equation that defines this transform is the following:

$$X(z) = \sum_{-\infty}^{\infty} x[n]z^{-n}$$ where $z = re^{j\omega}$

We know beforehand that the DT Fourier Transform H(z), uses values of z that lie on the UNIT CIRCLE of that plane (hence why the DTFT repeats itself at every 2π). The Z-Transform allows us to generalize z to any point on the plane thanks to including the radius term in the expression of z.

Unlike the Laplace Transform which focuses on the Cartesian coordinates, the Z-Transform is oriented towards using polar coordinates.



Region of Convergence and Its Properties

Similar to the Laplace Transform, the Z-Transform might not converge for every possible value of z. Hence, there exists certain regions of convergence for some Z-Transforms.

Unlike the Laplace Transform's rectangular sweeping region of convergence, the regions of convergence for Z-Transforms are disc-shaped.

The boundaries for this region are defined by the poles. Right-sided signals have a region of convergence that starts at the outermost pole and expands OUTWARDS indefinitely. Left-sided signals have a region of convergence that starts at the innermost pole and expands INWARDS towards the origin (and may or may not include the origin). Signals that are composed of both right-sided and left-sided signals have a region of convergence that resembles a ring.

Recalling the Laplace Transform, a system is said to be stable if the Laplace Transform of its impulse response includes the imaginary axis. In the case of Z-Transforms, the region of convergence must include the UNIT-CIRCLE for stable systems.

Inverse Z-Transform

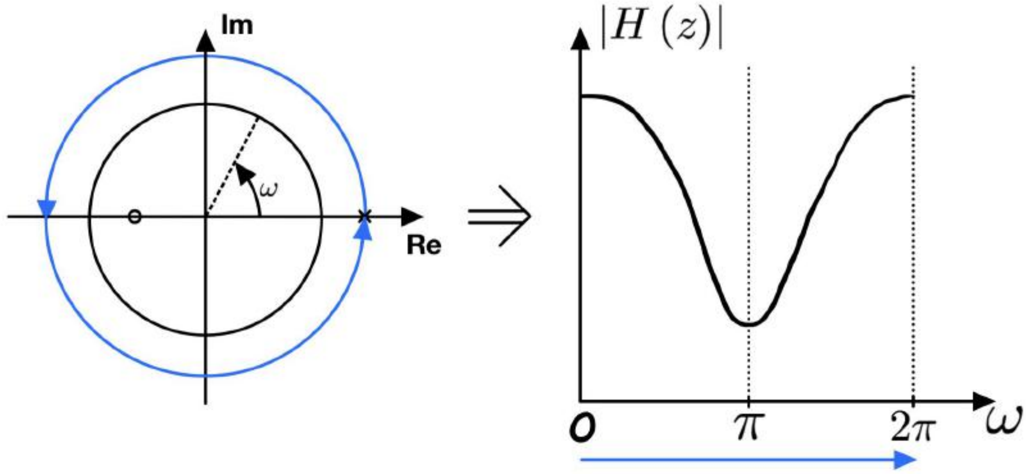
The inverse Z-Transform is defined by the following complex integral:

$$x[n] = \frac{1}{2\pi j}\oint_C X(z)z^{n-1}dz$$

Solving the integral is out of the scope of BIEN350 and hence the Z-Transform pairs table must be used to find inverse Z-Transforms. In most cases, the given Z-Transform in an exercise is too complicated to find in the table and must thus be broken down into simpler terms and then find the inverse Z-Transform of each. Many mathematical tricks come in handy in these cases (e.g. partial fraction decomposition).

Finding the Magnitude Response from the Pole-Zero Map

Just like in the case of the Laplace Transform, the Pole-Zero map of the Z-Transform can be used to sketch the system's magnitude response. The procedure is essentially identical, however, instead of starting at the origin and moving upward along the imaginary axis, start at $\omega = 0$ and move counterclockwise around the unit circle! Remember that for DT signals, the magnitude response repeats itself every 2π. Note that the symmetry around π portrayed below is a consequence of the poles and zeros only existing on the real axis in this specific example. Such symmetry is *not* guaranteed.



Properties of Z-transform

Section Reference	Sequence	Transform	ROC
	$x[n]$	$X(z)$	R_x $r_R < \|z\| < r_L$
	$x_1[n]$	$X_1(z)$	R_{x_1}
	$x_2[n]$	$X_2(z)$	R_{x_2}
3.4.1	$ax_1[n] + bx_2[n]$	$aX_1(z) + bX_2(z)$	Contains $R_{x_1} \cap R_{x_2}$
3.4.2	$x[n - n_0]$	$z^{-n_0}X(z)$	R_x, except for the possible addition or deletion of the origin or ∞
3.4.3	$z_0^n x[n]$	$X(z/z_0)$	$\|z_0\|R_x$ $\|z_0\|r_R < \|z\| < \|z_0\|r_L$
3.4.4	$nx[n]$	$-z\dfrac{dX(z)}{dz}$	R_x, except for the possible addition or deletion of the origin or ∞
3.4.5	$x^*[n]$	$X^*(z^*)$	R_x
	$\mathcal{R}e\{x[n]\}$	$\frac{1}{2}[X(z) + X^*(z^*)]$	Contains R_x
	$\mathcal{J}m\{x[n]\}$	$\frac{1}{2j}[X(z) - X^*(z^*)]$	Contains R_x
3.4.6	$x^*[-n]$	$X^*(1/z^*)$	$1/R_x$ $1/r_L < \|z\| < 1/r_R$
	$x[n]$ is real $x[-n]$	$X(1/z)$	
3.4.7	$x_1[n] * x_2[n]$	$X_1(z)X_2(z)$	Contains $R_{x_1} \cap R_{x_2}$
3.4.8	Initial-value theorem:		
	$x[n] = 0, \quad n < 0$ $\quad \lim_{z\to\infty} X(z) = x[0]$		

Unilateral Z-transform

The unilateral Z-Transform is defined by the following equation:

$$X(z) = \sum_{0}^{\infty} x[n]z^{-n}$$

This variant of the Z-Transform, along with the time-shift property are extremely helpful in solving Difference Equations:

$$x[n] \rightarrow X(z)$$

$$x[n-1] \rightarrow x[-1] + z^{-1}X(z)$$

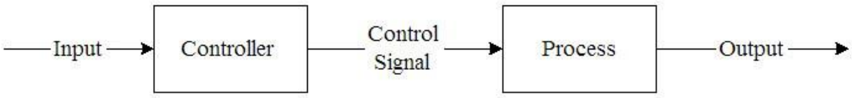
$$x[n-k] = x[-k] + z^{-1}x[-k+1]\ldots z^{-k+1}x[-1] + z^{-k}X(z)$$

When solving DE’s using the Z-Transform, the same approach can be used as the one for solving ODES with Laplace Transforms. The only difference here is the type of transform being used.

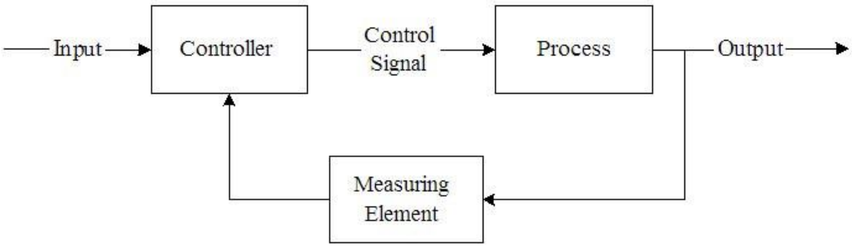
Control Systems in General

Control systems regulate the function of certain devices and systems using control-loops. The aim of a control system is to achieve the desired output of a system despite any external and internal effects that impede it from achieving this. In most cases, control systems are used to turn unstable systems into stable ones.

Open Loop vs. Closed Loop Systems



Open Loop System



Closed Loop System

https://medium.com/@mustafamlokhandwala/open-loop-idiocy-vs-closed-loop-intelligence-256f6260763f

Open loop control does not involve any FEEDBACK from the output signal while closed loop control features the feedback of the output signal to the control module. In other words, a closed loop system has the ability to self-correct.

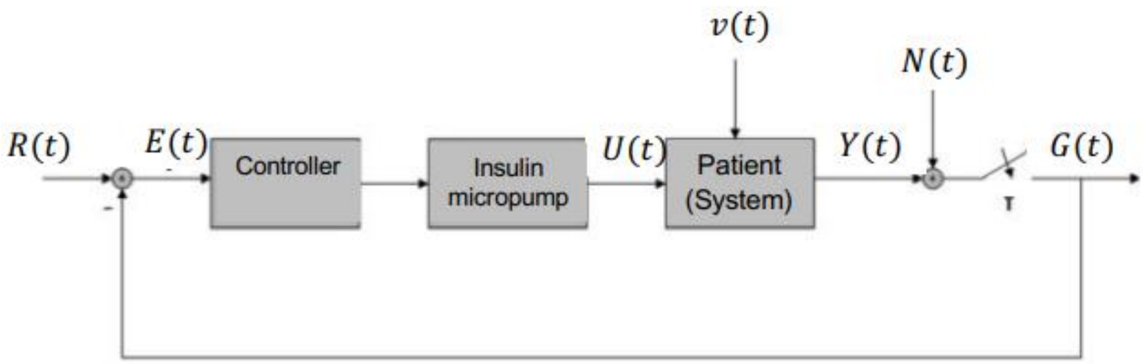
Glucose Regulation Example

Normally, hormones such as insulin and glucagon regulate the blood sugar level in the blood thanks to a natural built-in control system in the body.

In the case of Type I Diabetes, the patient suffers from insulin deficiency, which renders the natural blood sugar control system weak in some cases, hence the need for external intervention.

One solution is using manual insulin injections, which resembles open-control, as the patient injects themselves with a certain amount of insulin beforehand and expects the blood glucose levels to drop.

A more effective solution is the insulin pump, which is a closed-loop controller that gathers feedback in the form of measuring glucose levels in the blood. Based on those levels and the target level of the patient, the controller can calculate and deliver the amount of insulin needed in a continuous manner.



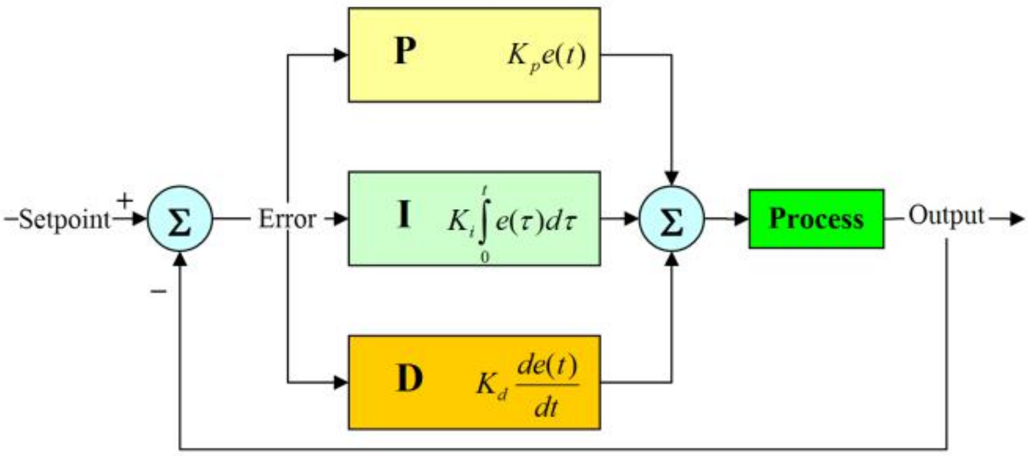
Meaning of each term:
- R(t): The target blood glucose level that the patient needs to reach (determined by the physician.
- G(t): The measured glucose level of the patient's blood.
- E(t): Simply the difference between the target and measured glucose levels, also known as the error term.
- U(t): The control signal generated by the insulin pump (i.e. the amount of insulin added)
- v(t): The patient's natural built-in natural glucose regulation system. The input to this system is the amount of insulin (and other factors as well), and the output from this system is the real glucose concentration in the blood. This is a highly complicated system and is often thought of as a 'black box'. The aim of the insulin pump is to control and stabilize this system.
- Y(t): The true value of the glucose concentration in the blood.
- N(t): All the noise that causes the discrepancy between the true value of the glucose concentration and the measurement. This is mainly due to instrumentation error.
- The switch: The insulin pump has the option of being on standby and not delivering any insulin (e.g. if the patient's blood sugar is already low)

Main limitation of this control system: The pump only uses insulin, which only lowers blood sugar, meaning it works in only 1 direction. If the patient experiences a sudden drop in glucose levels below the normal benchmark, the insulin pump will not be able to help in any way. Using a hormone that counters the effect of insulin (glucagon) will make this controller better.

Proportional-Integral-Derivative Control (PID)

A PID Controller is a type of controller that uses information on the following to regulate the system:
1) The magnitude of the error term (P)
2) The rate at which this error term changes (D)
3) The accumulation of this error term over time (I)

Each one of the mentioned terms has a controller coefficient (k_p. k_d, and k_i). These coefficients are design parameters that can be manipulated when tuning the controller.



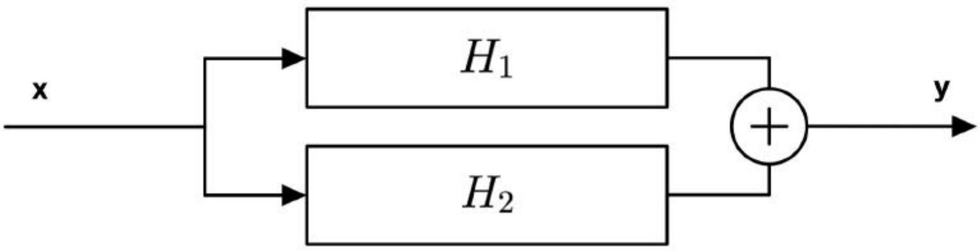
Introduction to Block Diagrams

Block diagrams are graphical representations of systems. They can represent, for example, parallel, series, and negative feedback systems. Note that the diagrams are extremely similar in the continuous and discrete time cases. The signals simply differ by the independent variable: t in the continuous case, and n in the discrete case. In the frequency domain, the independent variable is s in the continuous case and z in the discrete case.

For parallel system block diagrams, the frequency responses of the individual elements combine as follows:

$H(s) = H_1(s) + H_2(s)$ (continuous time)

$H[z] = H_1[z] + H_2[z]$ (discrete time)



For series system block diagrams, the frequency responses of the individual elements combine as follows:

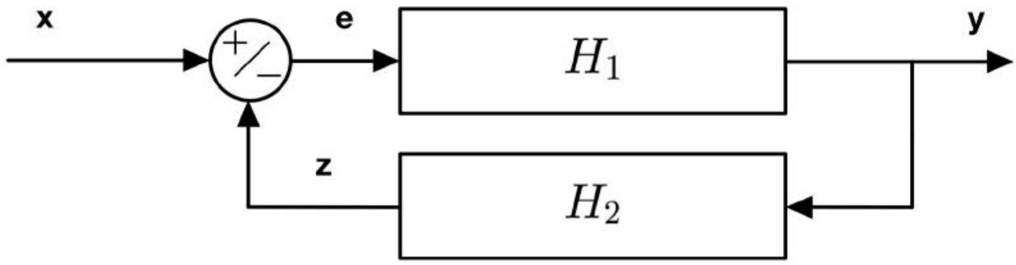
$H(s) = H_1(s)H_2(s)$ (continuous time)

$H[z] = H_1[z]H_2[z]$ (discrete time)

For negative feedback block diagrams, the frequency responses of the individual elements combine as follows:

$$H(s) = \frac{H_1(s)}{1 + H_1(s)H_2(s)}$$

$$H[z] = \frac{H_1[z]}{1 + H_1[z]H_2[z]}$$

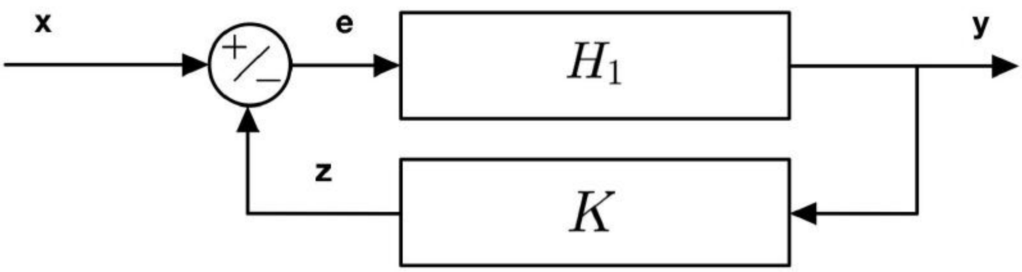


Note that in the negative feedback case, the product $H_1 H_2$ is called the *loop gain*.

Block diagrams (Laplace, Z) with HG (loop gain)

For the purposes of this course, we will assume all systems are CAUSAL. That means we only need to worry about regions of convergence that are right-hand side planes (in the continuous time case) or circular regions tending to infinity (in the discrete time case).

The goal of feedback is the stabilization of the system. The H_2 term can be tweaked, changing the loop gain. Choosing an appropriate H_2 will stabilize the system. Consider the case where $H_2 = K$, with K, a constant.



The system, $H(s) = \frac{b}{s - a}$, becomes $H(s) = \frac{b}{s - a + Kb}$.

While the original system is not always stable, the feedback system is stable for $K > \frac{a}{b}$. All we need to do is pick an appropriate value of K!

Then, consider the following system:

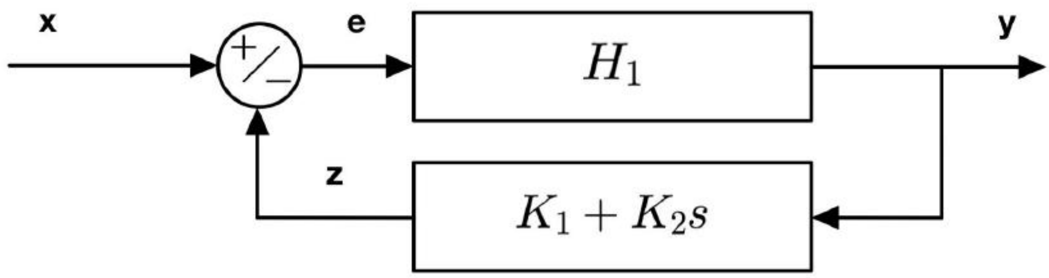
$$H_1(s) = \frac{b}{s^2 + a}$$

If we used $H_2(s) = K$ as was the case above, we would get the following corrected system:

$$H(s) = \frac{b}{s^2 + (a + Kb)}$$

The poles for this new transfer function are the roots of $s^2 + (a + Kb)$, which must be positive. Given that the system is causal, the ROC would not include the real axis which implies that the system cannot be stabilized by a zeroth-order loop branch $H_2(s) = K$.

Instead, we should use $H_2(s) = K_1 + K_2 s$,



The new transfer function would emerge as:

$$H(s) = \frac{b}{s^2 + bK_2 s + (a + bK_1)}$$

The roots of the denominator for H(s) would be

$$p1, p2 = \frac{-bK_2 \pm \sqrt{b^2 K_2^2 - 4(a + bK_1)}}{2}$$

Hence, it is possible for both of the roots of the polynomial to be negative given the appropriate K_1 and K_2 values, which means that we need a first order feedback branch to stabilize this second order system.

When we vary the value of K, we are actually moving the locations of the system's poles. The system is stabilized once we choose a value of H_2 such that the poles are all negative on the real axis (in the continuous case) or within the unit circle (in the discrete case).

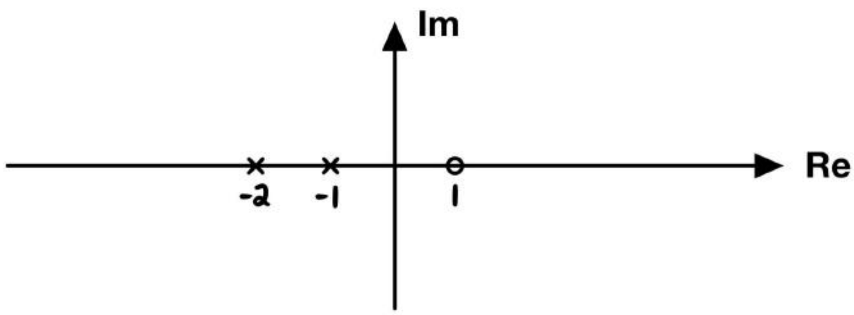
Root Locus Method

The Root Locus method allows us to evaluate the behaviour of closed-loop systems at various values of K. Remember these key steps to handle any root locus problem:

1) Find G(s)H(s), where H(s) is the transfer function of the original system and G(s) is that of the feedback loop.

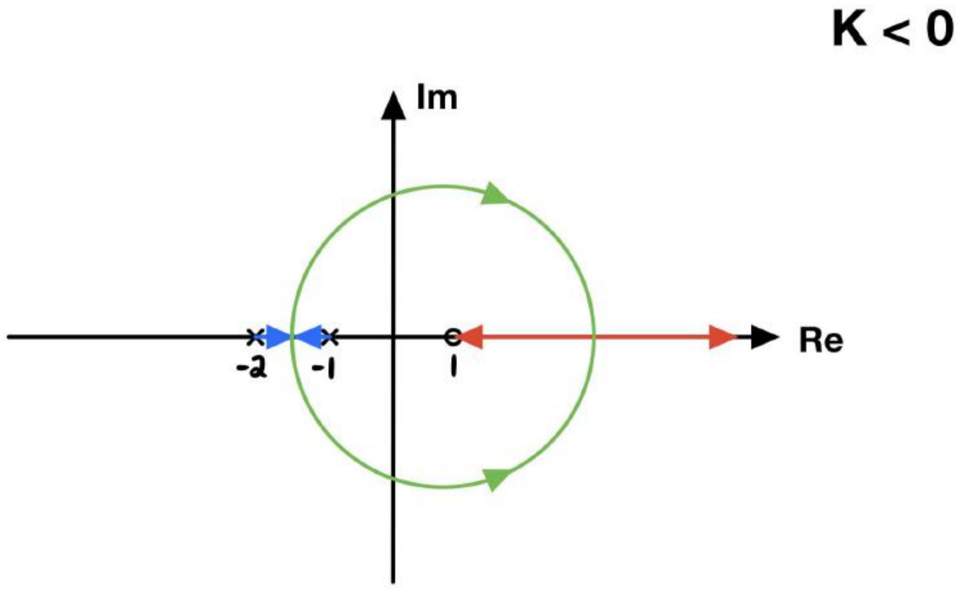
   Example: $G(s)H(s) = \frac{s - 1}{(s + 1)(s + 2)}$

2) Mark the poles and zeros of G(s)H(s) on the real-imaginary plane.

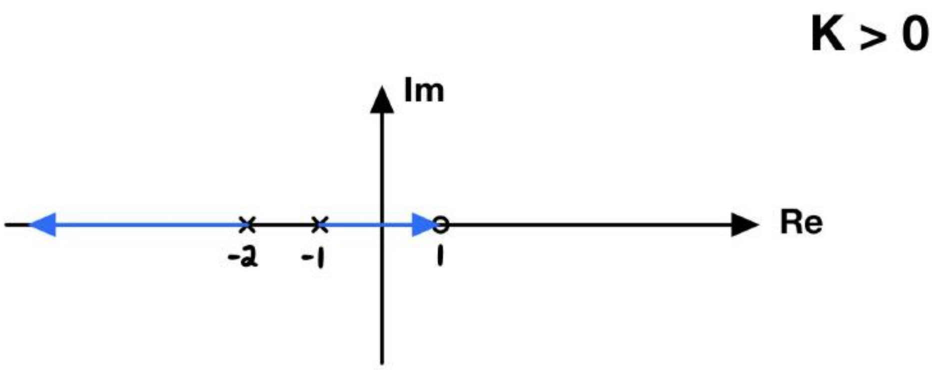


3) Start at poles $(K = 0)$

4) End at zeroes. If there are no remaining zeros, go to infinity. $(|K| = \infty)$
5) For $K < 0$, look for segments of the real axis to the left of an odd number zeroes and poles. Once the 2 traces come into contact, the 2 traces would both leave the real axis and head either towards the zeros or infinity in a symmetric pattern. Note that for the scope of BIEN 350, these diversions will be either in the shape of a circle that loops back to the real axis, or vertical lines that go to $\omega = \pm\infty$



6) Perform the same procedure for $K > 0$



7) Using the criteria for system stability, determine K values for which the system is stable. In other words, look for all the values of K when all the traces are inside the unit circle for DT systems or in the negative left half-plane

<u>Note</u>: Each zero can only accept 1 trace coming from a pole.