

ashapes,arrows

Introduction to Signals, Systems, and Feedback Control

Massachusetts Institute of Technology, January 2019

Rhian Chavez

Lila Wine

Contents

1	Linear Time Invariant Systems	4
1.1	Discrete LTI Systems Intro	4
1.1.1	Getting Comfortable with Discrete Signals	4
1.1.2	Discrete Systems	5
1.1.3	Discrete Feedback & Feedforward Systems	6
1.1.4	System Functional	7
1.1.5	Combinations	7
1.2	Continuous LTI Systems	8
1.2.1	Important Continuous Inputs	8
1.3	Impulse Response	9
1.3.1	Discrete	9
1.3.2	Continuous	9
1.4	Convolution & General LTI System Response	10
1.4.1	Discrete	10
1.4.2	Continuous	11
1.4.3	More on Convolution	11
2	Frequency Domain	12
2.1	Review of Complex Numbers	12
2.2	Continuous Fourier Series	13
2.2.1	Real Representation	13
2.2.2	Complex Representation	14
2.3	Discrete Fourier Transform	15
2.4	Continuous Fourier Transform	16
2.5	Discrete Time Fourier Transform	17
3	Relationship between Convolution & Multiplication	19
3.1	Aperiodic Continuous Functions	19
3.2	Aperiodic Discrete Functions	19
3.3	Periodic Continuous Functions	20
3.4	Periodic Discrete Functions	21
3.5	Application to LTI Systems	21
3.5.1	Discrete	21
3.5.2	Continuous	22
3.6	More on Circular Convolution	22
4	Simple Filtering & Bode Plots	25
4.1	Discrete Time	25
4.2	Continuous Time	26
4.3	Bode Plots	27
5	Ideal Sampling	29

5.1	Aliasing	29
5.2	Analog to Digital	30
5.3	Digital to Analog	33
6	Laplace & Z-Transforms, Feedback Control	35
6.1	Introduction	35
6.1.1	Unilateral Laplace Transform	35
6.1.2	Bilateral Z Transform	36
6.2	Convolution & Multiplication	37
6.2.1	Continuous	37
6.2.2	Discrete	38
6.3	Application to LTI Systems	38
6.3.1	Discrete Time Systems	38
6.3.2	Continuous Time Systems	40
7	PID Feedback Control	43
7.1	Root Locus	43
7.2	Routh-Hurwitz Criterion	45
7.3	PID Controller Design	47
8	Basic Signal Modulation	52
8.1	Amplitude Modulation	52
8.2	Frequency Modulation	53
8.3	Phase Modulation	54
8.3.1	Binary Phase Shift Keying	54
9	Other Applications	56
9.1	Spectrum Smearing & Windowing	56
9.1.1	Spectrum Smearing	56
9.1.2	Windowing	57
9.2	Fast Fourier Transform	59
9.3	Time-Frequency Uncertainty Principle	59

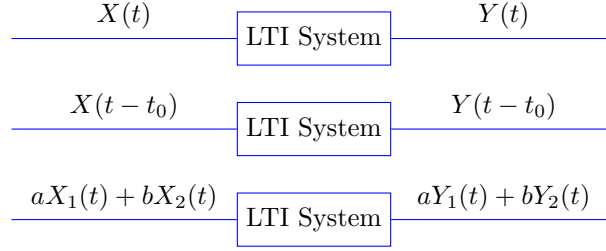
1 Linear Time Invariant Systems

Linear time invariant (LTI) systems consist of a **relationship between the input and output** of some linear function which is independent of absolute time. For example, the two-dimensional continuous function

$$y = 10x + 3 \quad (1.1)$$

can be thought of as a system which takes an input x and returns an output y . More generally, we can refer to the output of some system as Y and the input of some system as X . In this course, we will analyze both continuous and discrete time systems.

LTI systems are defined by the following characteristics:



There are many other examples of LTI systems such as; derivative, integral, etc.

To emphasize the more generalized concept of an LTI system, we will begin with discrete time systems.

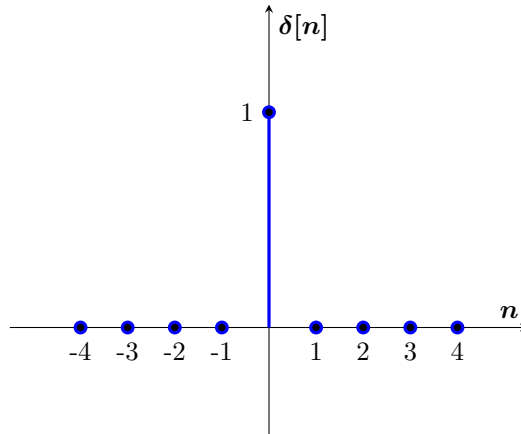
1.1 Discrete LTI Systems Intro

1.1.1 Getting Comfortable with Discrete Signals

We will represent the input to a LTI system as a discrete function of sample number $x[n]$ where n is the sample number. For example, the *unit sample function*, written as $\delta[n]$, is defined by the following relationship:

$$\delta[n] \equiv \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad (1.2)$$

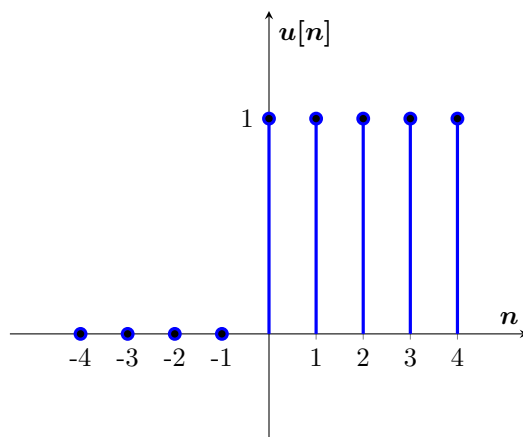
The unit sample function can be represented graphically by the following discrete plot:



Another discrete function that we will reference many times throughout the course is known as the *unit step function*, written as $u[n]$.

$$u[n] \equiv \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases} \quad (1.3)$$

The unit sample function can be represented graphically by the following discrete plot:



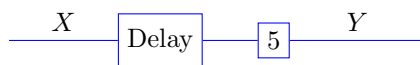
Now that we are familiar with a couple types of discrete signals, we are ready to think about using them as inputs to discrete systems.

1.1.2 Discrete Systems

There are a few different ways we are going to represent discrete systems; functional, graphical, and operational. For example, imagine the a function that takes an input X , and returns 5 times the input as Y one time step later. The functional representation of that system would be the following:

$$y[n] = 5x[n - 1] \quad (1.4)$$

The graphical representation of such a function would be the following:

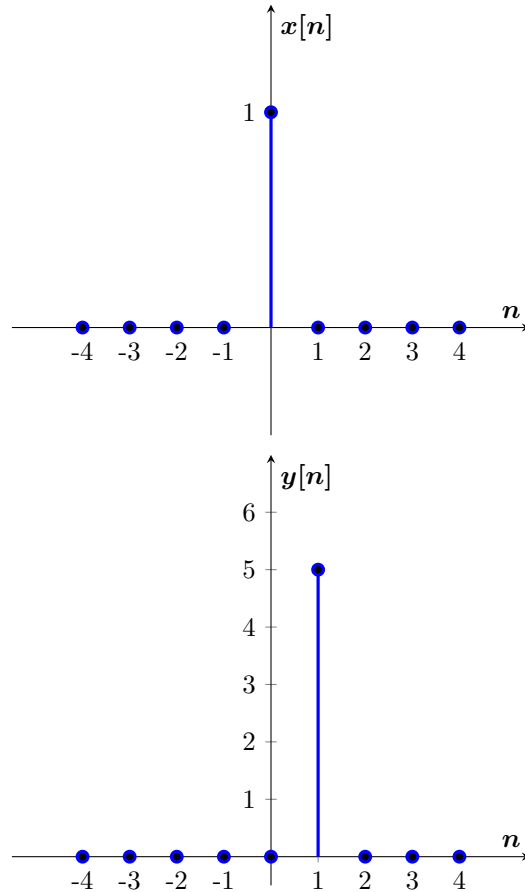


Where the triangle shape represents the constant scale factor of 5, and the block labeled "Delay" represents the single sample delay. Finally, the operator representation of the previous signal is simply:

$$Y = 5Z^{-1}X \quad (1.5)$$

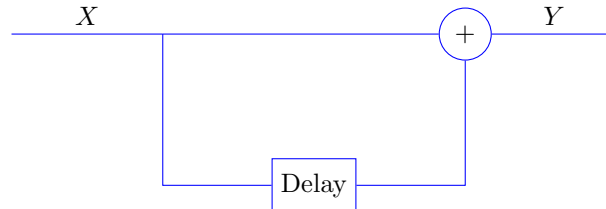
Where the Z^{-1} represents the single sample delay, (the reason for this will become clear later in the course). We simply abstract away the complexity of the input and output and represent them compactly as X and Y . We will use the operator Z^{-1} to represent a single sample delay, therefore if the input were instead delayed by 5 samples, the operator would then be Z^{-5} . **It is extremely important to remember that the operators (delay and scale) follow the algebraic distributive and commutative laws.**

If we were to apply this system to the unit sample signal, the input $x[n]$ and output $y[n]$ would be the following:



1.1.3 Discrete Feedback & Feedforward Systems

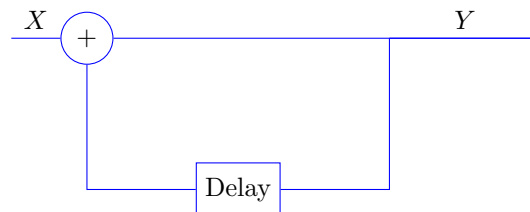
We can represent a **feedforward** system graphically as the following:



The input X at a time-step n is summed with the input with at time-step $n - 1$ and then returned as the output Y . We represent this in functional and operator notation as the following:

$$y[n] = x[n] + x[n - 1] \iff Y = X + Z^{-1}X = (1 + Z^{-1})X \quad (1.6)$$

We can represent a **feedback** system graphically as the following:



The input X at a time-step n is summed with the previous output at time-step $n - 1$ and then returned

as the output Y . We represent this in functional and operator notation as the following:

$$y[n] = x[n] + y[n-1] \iff Y = X + Z^{-1}Y \iff (1 - Z^{-1})Y = X \quad (1.7)$$

1.1.4 System Functional

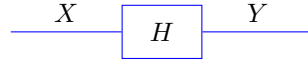
A very useful way to abstract the relationship between the input signal X and output signal Y of a system is known as the *system functional*. When a discrete system is written in operator notation, the most general system (that doesn't look forward in time) is the following:

$$\sum_{i=0}^{\infty} a_i Z^{-i} Y = \sum_{j=0}^{\infty} b_j Z^{-j} X \quad (1.8)$$

This equation can be thought of as all possible feedforward and feedback relations. The j^{th} value on the right hand side represents a scaled delayed feedforward relation, each i^{th} value on the left hand side represents a scaled delayed feedback relation. We define the system functional as the following:

$$H = \frac{Y}{X} = \frac{\sum_{j=0}^{\infty} b_j Z^{-j}}{\sum_{i=0}^{\infty} a_i Z^{-i}} \quad (1.9)$$

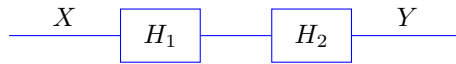
System functionals are very useful because they can abstract away the relationship between input X and output Y into a single operator H . In general,



$$Y = HX \quad (1.10)$$

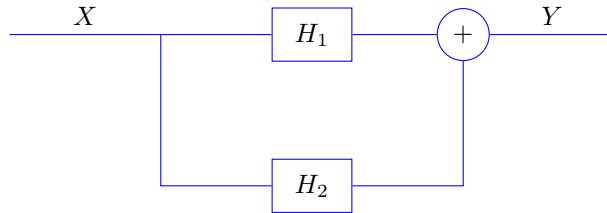
1.1.5 Combinations

Now that we have a simple relationship between inputs and outputs of systems, we can extrapolate to sequential, forward additional, and backward additive combinations. Given a system with input X and output $H_1 X$, if we use the output of the first system as the input to the second system H_2 , the final output will be simply $H_2 H_1 X$. **Since the operator notation of systems is commutative, the order of the system functionals does NOT matter.** Graphical representation below:



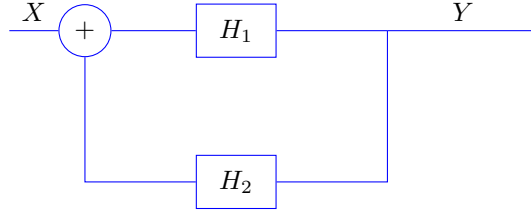
$$Y = H_2 H_1 X \iff \frac{Y}{X} = H = H_2 H_1 \quad (1.11)$$

Forward additive combinations can be abstracted from the simple feedforward system discussed in section 1.1.3. We simply replace the delay operator with a functional operator H_2 and place an operator H_1 in the unscaled path.



$$Y = (H_1 + H_2)X \iff \frac{Y}{X} = H = H_1 + H_2 \quad (1.12)$$

Backward additive combinations can be abstracted from the simple feedback system discussed in section 1.1.3. We simply replace the delay operator with a functional operator H_2 and place an operator H_1 in the unscaled path.



$$Y = H_1X + H_1H_2Y \iff \frac{Y}{X} = H = \frac{H_1}{1 - H_1H_2} \quad (1.13)$$

We should now be able to find the system functional of every possible combination of systems by collapsing smaller system functionals until we are left with only one.

1.2 Continuous LTI Systems

Continuous LTI systems are very similar to discrete LTI systems, they represent the relationship between a continuous input signal and continuous output signal. Linear differential equations are a good example of a continuous LTI system, for example the second order differential equation:

$$f''(x) + af'(x) + bf(x) = g(x) \quad (1.14)$$

has an input $g(x)$ and output $f(x)$.

1.2.1 Important Continuous Inputs

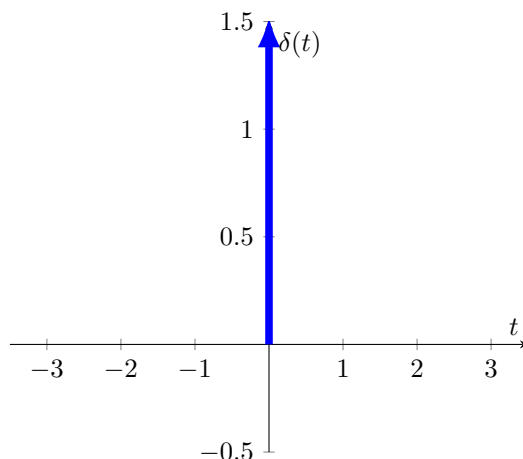
The continuous analog of the discrete unit sample function is known as a *delta function*. The delta function is defined by the following relationships:

$$\delta(t) \equiv \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases} \quad (1.15)$$

$$\int_{-\infty}^{\infty} dt f(t) \delta(t - t_0) = f(t_0) \quad (1.16)$$

It is important to recognize that integral over $\delta(t)$ is equal to 1, (the area under the function is 1). The delta function also seems to *pick out* the value of the $f(t)$ at t_0 .

A plot of $\delta(t)$ below:



We will not go into depth regarding the formulation of the delta function.

The second important continuous input is known as Heaviside step function, or *unit step function*. The unit step function is defined by the following relationship:

$$u(t) \equiv \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (1.17)$$

It is important to note that the derivative of $u(t)$ is $\delta(t)$! You should be able to convince yourself that this is true.

1.3 Impulse Response

1.3.1 Discrete

The impulse response of a system is the response to a unit impulse signal $\delta[n]$ (discrete time) or a delta function $\delta(t)$ (continuous time). We denote the impulse response of a system as $h[n]$

For example: Given a discrete system described by the function:

$$y[x] = 3x[n] - 2x[n - 1] \quad (1.18)$$

Plugging in $\delta[n]$ for $x[n]$, (setting the input as the unit impulse signal), we get:

$$h[x] = 3\delta[n] - 2\delta[n - 1] \quad (1.19)$$

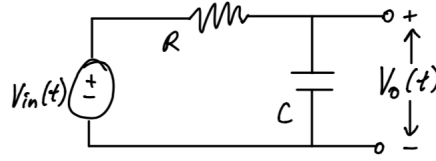
We can represent the output as an array where index n corresponds to $h[n]$ as the following:

`[3, -2, 0, 0, 0, 0, ...]`

As you can possibly imagine, this is an extremely simple system, and systems with complex feedback loops will have much more complicated impulse responses. We will learn how to solve for the general impulse response later in the course.

1.3.2 Continuous

Continuous systems also have an impulse response. We denote the impulse response of a continuous system as $h(t)$. A very applicable example is the impulse response of an RC circuit.



Suppose we have a simple RC circuit as shown above, how can we find the impulse response, (to voltage input $\delta(t)$). Well, we can take advantage of the fact that, for LTI systems, **the derivative of the unit step response is the impulse response**. This will be proved later in the course.

The differential equation governing the behavior of a driven RC circuit is the following:

$$\dot{V}_0(t) + \frac{V_0(t)}{RC} = \frac{V_{in}(t)}{RC} \quad (1.20)$$

If we set $V_{in}(t) = u(t)$, we can solve the following equation which applies for $t > 0$ and apply the initial condition of $V_0(t = 0) = 0$.

$$\dot{V}_0(t) + \frac{V_0(t)}{RC} = \frac{1}{RC} \quad (1.21)$$

The solution to this differential equation is the following:

$$V_o(t) = u(t)(1 - e^{-t/RC}) \quad (1.22)$$

Now we simply differentiate this unit step response to find the impulse response! It is important to remember that $u(t)$ is a function, with a derivative $\delta(t)$.

$$\hat{D}_t V_o(t) = \delta(t) - \delta(t)e^{-t/RC} + \frac{u(t)}{RC}e^{-t/RC} = \frac{u(t)}{RC}e^{-t/RC} = h(t) \quad (1.23)$$

The $\hat{D}_t$ operator represents differentiation with respect to time, and the $u(t)$ is necessary because the solution to both the unit step input and delta function input is only non-zero for $t \geq 0$.

1.4 Convolution & General LTI System Response

1.4.1 Discrete

Suppose you have a signal $x[n]$ defined by the following array where index n corresponds to $x[n]$:

[3, 33, 333, 3333, 33333, 333333]

We could also represent this system in functional notation by the following:

$$x[n] = 3\delta[n] + 33\delta[n-1] + 333\delta[n-2] + 3333\delta[n-3] + 33333\delta[n-4] + 333333\delta[n-5] \quad (1.24)$$

By the same logic, **any** signal can be represented by a sum of shifted and scaled delta functions. Now, **if we know the the response of an LTI system to a single delta function, we know the response of the system to any signal!** Since the system is linear and time invariant, all we have to do is scale and shift the impulse response appropriately. The logic is as follows:

$$\delta[n] \rightarrow h[n] \quad (1.25)$$

$$\delta[n-k] \rightarrow h[n-k] \quad (1.26)$$

$$x[k]\delta[n-k] \rightarrow x[k]h[n-k] \quad (1.27)$$

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \rightarrow y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \quad (1.28)$$

The sum performed in line 28 is defined as the *convolution* of $x[n]$ and $h[n]$ and is commutative.

1.4.2 Continuous

Suppose you have a function $x(t)$ defining an input signal (ex: $u(t) \cos t$). By the definition of the $\delta(t)$, we could also represent $x(t)$ by the following expression:

$$x(t) = \int_{-\infty}^{\infty} d\tau x(\tau) \delta(t - \tau) \quad (1.29)$$

By the same logic, **any** continuous signal can be represented by a an integral of shifted and scaled delta functions. Now, **if we know the the response of an LTI system to a single delta function, we know the response of the system to any signal!** Since the system is linear and time invariant, all we have to do is scale and shift the impulse response appropriately. The logic is as follows:

$$\delta(t) \rightarrow h(t) \quad (1.30)$$

$$\delta(t - \tau) \rightarrow h(t - \tau) \quad (1.31)$$

$$x(\tau) \delta(t - \tau) \rightarrow x(\tau) h(t - \tau) \quad (1.32)$$

$$x(t) = \int_{-\infty}^{\infty} d\tau x(\tau) \delta(t - \tau) \rightarrow y(t) = \int_{-\infty}^{\infty} d\tau x(\tau) h(t - \tau) \quad (1.33)$$

The integral performed in line 33 is defined as the *convolution* of $x(t)$ and $h(t)$ and is commutative.

1.4.3 More on Convolution

Convolution is defined by the following equations for discrete and continuous systems, and denoted by the $*$ operator.

$$x[n] * y[n] = \sum_{k=-\infty}^{\infty} x[k] y[n - k] \quad (1.34)$$

$$x(t) * y(t) = \int_{-\infty}^{\infty} d\tau x(\tau) y(t - \tau) \quad (1.35)$$

A graphical representation of convolution can be found [here](#). We will discuss how to find the convolution of simple discrete signals graphically in class.

2 Frequency Domain

2.1 Review of Complex Numbers

Complex numbers are usually denoted by the following:

$$z = a + jb \quad \text{where } a, b \in \mathbb{R}, \quad j \equiv \sqrt{-1} \quad (2.1)$$

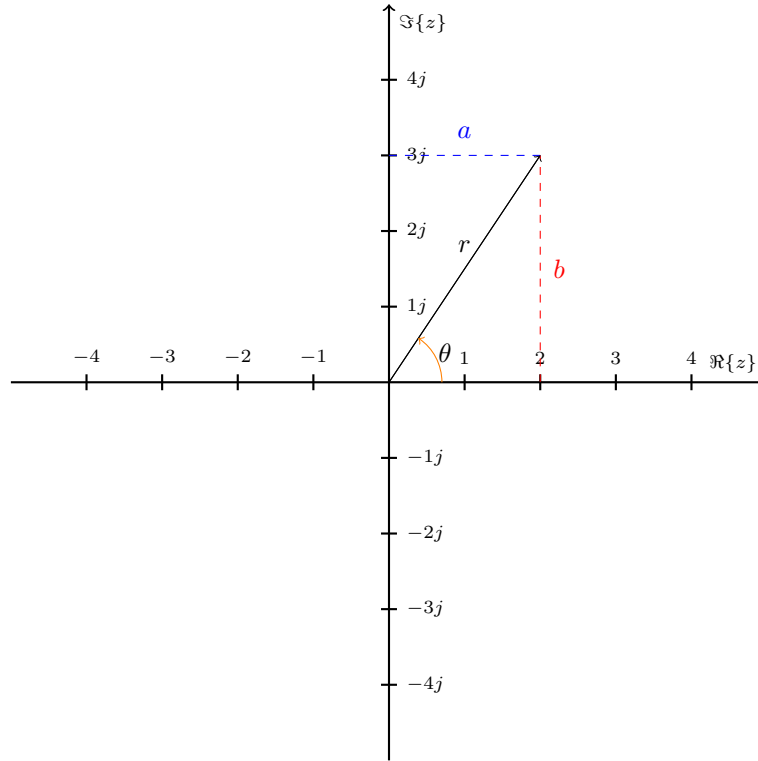
We will almost always use Euler's formula to describe and work with complex numbers.

$$\boxed{e^{j\theta} = \cos \theta + j \sin \theta} \quad (2.2)$$

This equation can be shown to be true by taking the Taylor expansion of $e^{j\theta}$ and comparing it with the well known Taylor expansions of $\cos \theta$ and $\sin \theta$.

$$e^{j\theta} = \sum_{n=0}^{\infty} \frac{(j\theta)^n}{n!} = \sum_{m=0}^{\infty} \frac{(j\theta)^{2m}}{(2m)!} + \sum_{k=0}^{\infty} \frac{(j\theta)^{2k+1}}{(2k+1)!} = \underbrace{\sum_{m=0}^{\infty} \frac{(-1)^m \theta^{2m}}{(2m)!}}_{\cos \theta} + j \underbrace{\sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k+1}}{(2k+1)!}}_{\sin \theta} \quad (2.3)$$

It is easiest to interpret this relationship by visualizing a complex number as a vector in the complex plane.



We will denote $r, \theta \in \mathbb{R}$ as the *magnitude* and *phase* of a complex number, respectively. Complex numbers will generally be referred to in the form $re^{j\theta}$. We will also often refer to the *complex conjugate*, which is simply $re^{-j\theta}$, (if the complex number is not in Euler form, **every** j must be changed to $-j$ to conjugate). You can always find the magnitude squared of a complex number by multiplying the complex number z by its complex conjugate. Interpreting a complex expression geometrically will prove extremely useful when finding r and θ .

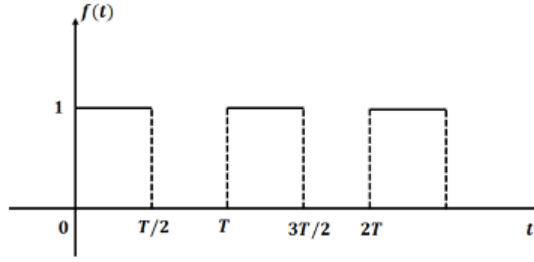
We will need to know how to use one more tool in this course, the Kronecker delta, δ_{ij} , defined by the following relationship:

$$\delta_{ij} \equiv \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \quad (2.4)$$

2.2 Continuous Fourier Series

2.2.1 Real Representation

Fourier Series are an extremely useful tool in engineering, physics, and mathematics. The theory stems from the idea that any periodic function can be constructed from a sum of weighted harmonics (sines and cosines) of the same periodicity. For example, say you have a square wave with period T on the real number line. Such as the following:



Fourier theory proposes that this square wave can be reconstructed from a sum of weighted sines and cosines of the same periodicity, so the following must be true:

$$f(t) = \sum_{k=0}^{\infty} a_k \cos(2\pi kt/T) + \sum_{k=1}^{\infty} b_k \sin(2\pi kt/T) \quad (2.5)$$

Note that although all of the harmonics don't have T as their minimum period, they are all still periodic with period T . Also note that the second sum starts k at index 1 instead of 0 because $\sin(0)$ is 0.

So how can we find the coefficients a_k and b_k ? We can take advantage of the orthogonality of all of the harmonics! By orthogonality of the harmonics I mean the following (where a and b are integers):

$$\int_{-T/2}^{T/2} dt \cos(2\pi at/T) \cos(2\pi bt/T) = \delta_{ab} \frac{T}{2} \quad (2.6)$$

$$\int_{-T/2}^{T/2} dt \sin(2\pi at/T) \sin(2\pi bt/T) = \delta_{ab} \frac{T}{2} \quad (2.7)$$

$$\int_{-T/2}^{T/2} dt \cos(2\pi at/T) \sin(2\pi bt/T) = 0 \quad (2.8)$$

We can find any a_n by the following method:

$$f(t) = \sum_{k=0}^{\infty} a_k \cos(2\pi kt/T) + \sum_{k=1}^{\infty} b_k \sin(2\pi kt/T) \quad (2.9)$$

$$f(t) \cos(2\pi nt/T) = \sum_{k=0}^{\infty} a_k \cos(2\pi kt/T) \cos(2\pi nt/T) + \sum_{k=1}^{\infty} b_k \sin(2\pi kt/T) \cos(2\pi nt/T) \quad (2.10)$$

$$\int_{-T/2}^{T/2} dt f(t) \cos(2\pi nt/T) = \int_{-T/2}^{T/2} dt \left(\sum_{k=0}^{\infty} a_k \cos(2\pi kt/T) \cos(2\pi nt/T) + \sum_{k=1}^{\infty} b_k \sin(2\pi kt/T) \cos(2\pi nt/T) \right) \quad (2.11)$$

$$\int_{-T/2}^{T/2} dt f(t) \cos(2\pi nt/T) = \sum_{k=0}^{\infty} \frac{T}{2} a_k \delta_{kn} \quad (2.12)$$

$$\int_{-T/2}^{T/2} dt f(t) \cos(2\pi nt/T) = \frac{T}{2} a_n \quad (2.13)$$

$$\boxed{a_n = \frac{2}{T} \int_{-T/2}^{T/2} dt f(t) \cos(2\pi nt/T)} \quad (2.14)$$

By the same logic, we can find any b_m by the following method:

$$f(t) = \sum_{k=0}^{\infty} a_k \cos(2\pi kt/T) + \sum_{k=1}^{\infty} b_k \sin(2\pi kt/T) \quad (2.15)$$

$$f(t) \sin(2\pi mt/T) = \sum_{k=0}^{\infty} a_k \cos(2\pi kt/T) \sin(2\pi mt/T) + \sum_{k=1}^{\infty} b_k \sin(2\pi kt/T) \sin(2\pi mt/T) \quad (2.16)$$

$$\int_{-T/2}^{T/2} dt f(t) \cos(2\pi mt/T) = \int_{-T/2}^{T/2} dt \left(\sum_{k=0}^{\infty} a_k \cos(2\pi kt/T) \sin(2\pi mt/T) + \sum_{k=1}^{\infty} b_k \sin(2\pi kt/T) \sin(2\pi mt/T) \right) \quad (2.17)$$

$$\int_{-T/2}^{T/2} dt f(t) \sin(2\pi nt/T) = \sum_{k=1}^{\infty} \frac{T}{2} b_k \delta_{kn} \quad (2.18)$$

$$\int_{-T/2}^{T/2} dt f(t) \sin(2\pi mt/T) = \frac{T}{2} b_n \quad (2.19)$$

$$\boxed{b_m = \frac{2}{T} \int_{-T/2}^{T/2} dt f(t) \sin(2\pi mt/T)} \quad (2.20)$$

You can interact with the reconstruction process of square wave above using the online mathematical tool Desmos [here](#). With any finite sum Fourier series, the jump discontinuities of the function will not be perfect, you can read more about this phenomenon [here](#).

2.2.2 Complex Representation

In practice, it is very convenient to represent sines and cosines in complex form instead. Suppose we start with our function reconstruction formula:

$$f(t) = \sum_{k=0}^{\infty} a_k \cos(2\pi kt/T) + \sum_{k=1}^{\infty} b_k \sin(2\pi kt/T) \quad (2.21)$$

We would like to replace $\cos(x)$ and $\sin(x)$ with their complex representations:

$$\cos x = \frac{e^{jx} + e^{-jx}}{2} \quad \& \quad \sin x = \frac{e^{jx} - e^{-jx}}{2j} \quad (2.22)$$

If we plug these relationships into our reconstruction formula we get:

$$f(t) = \sum_{k=0}^{\infty} a_k \left(\frac{e^{j2\pi kt/T} + e^{-j2\pi kt/T}}{2} \right) + \sum_{k=1}^{\infty} b_k \left(\frac{e^{j2\pi kt/T} - e^{-j2\pi kt/T}}{2j} \right) \quad (2.23)$$

Grouping terms:

$$f(t) = \sum_{k=0}^{\infty} \frac{a_k - jb_k}{2} e^{j2\pi kt/T} + \frac{a_k + jb_k}{2} e^{-j2\pi kt/T} \quad (2.24)$$

Where we are allowed to group the sums because the second sum starting at 0 is inconsequential. This function can be re-written as:

$$f(t) = \sum_{k=0}^{\infty} \frac{a_k - jb_k}{2} e^{j2\pi kt/T} + \sum_{k=0}^{-\infty} \frac{a_{-k} + jb_{-k}}{2} e^{j2\pi kt/T} \quad (2.25)$$

And even further simplified as:

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j2\pi nt/T} \quad (2.26)$$

Where

$$c_n = \begin{cases} (a_n - jb_n)/2 & n > 0 \\ (a_{-n} + jb_{-n})/2 & n < 0 \\ a_0 & n = 0 \end{cases} \quad (2.27)$$

Fortunately, there is still a very simple way to calculate the c_n coefficients without using a_n and b_n . We must take advantage of the following fact:

$$\int_{-T/2}^{T/2} dt e^{j2\pi mt/T} = T\delta_{0,m} \quad m \in \mathbb{Z} \quad (2.28)$$

This can be shown to be true by expanding the complex exponential into cosine and sine. Therefore, by the same logic as we used in the real representation of Fourier Series, we can find any c_m :

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j2\pi nt/T} \quad (2.29)$$

$$f(t)e^{-j2\pi mt/T} = \sum_{n=-\infty}^{\infty} c_n e^{j2\pi nt/T} e^{-j2\pi mt/T} \quad (2.30)$$

$$\int_{-T/2}^{T/2} dt f(t)e^{-j2\pi mt/T} = \int_{-T/2}^{T/2} dt \left(\sum_{n=-\infty}^{\infty} c_n e^{j2\pi nt/T} e^{-j2\pi mt/T} \right) \quad (2.31)$$

$$\int_{-T/2}^{T/2} dt f(t)e^{-j2\pi mt/T} = \int_{-T/2}^{T/2} dt \sum_{n=-\infty}^{\infty} c_n e^{j2\pi(n-m)t/T} \quad (2.32)$$

$$\int_{-T/2}^{T/2} dt f(t)e^{-j2\pi mt/T} = \sum_{n=-\infty}^{\infty} T c_n \delta_{nm} \quad (2.33)$$

$$\int_{-T/2}^{T/2} dt f(t)e^{-j2\pi mt/T} = T c_m \quad (2.34)$$

$$c_m = \frac{1}{T} \int_{-T/2}^{T/2} dt f(t)e^{-j2\pi mt/T} \quad (2.35)$$

From now on, we will only use Fourier theory in the complex domain.

2.3 Discrete Fourier Transform

The discrete Fourier transform is the discrete analog of the continuous Fourier series. The theory stems from the fact that the vectors:

$$u_k = \left[e^{j2\pi kn/N} | n \in \{0, 1, \dots, N-1\} \right]^T \quad (2.36)$$

form an orthogonal basis over the set of N-dimensional complex vectors, (this is basically saying that sines and cosines of the same frequency k form a basis, but again, complex exponentials are much easier

to work with). This fact will not be proven in this course. Note that $u_k = u_{k+N}$; this is a key cyclic property of this basis.

If the vectors u_k form a basis for an n dimensional complex space, we should be able to construct any n dimensional vector from their linear superposition. Suppose we have an N dimensional vector $x[n]$ that is the input to one of our LTI systems discussed in section 1. We should be able to construct it as the following:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} \hat{X}[k] e^{j2\pi kn/N} \quad (2.37)$$

Where $\hat{X}[k]$ denotes the coefficients of our basis vectors, also known as the **Discrete Fourier Transform** of $x[n]$. The division by N is simply a scale necessary scale factor due to the nature of the construction of $\hat{X}[k]$. Note again that, given this definition, $x[n] = x[n+N]$; the discrete Fourier transform (DFT) assumes periodicity similar to the continuous Fourier series.

So how can we find $\hat{X}[k]$? We again take advantage of the orthogonality of our chosen basis vectors.

$$u_k^T u_{k'} = \sum_{n=0}^{N-1} e^{j2\pi kn/N} e^{-j2\pi k'n/N} = \sum_{n=0}^{N-1} e^{j2\pi(k-k')n/N} = N\delta_{k,k'} \quad (2.38)$$

This can be calculated explicitly, but it is sufficient to remember that $e^{j\theta} = \cos\theta + j\sin\theta$ and think about summing N evenly spaced points along one period of a sine wave; there is the same amount of positive function as there is negative! Given this information, we may now find any $\hat{X}[k']$ as follows:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} \hat{X}[k] e^{j2\pi kn/N} \quad (2.39)$$

$$x[n] e^{-j2\pi k'n/N} = \frac{1}{N} \sum_{k=0}^{N-1} \hat{X}[k] e^{j2\pi kn/N} e^{-j2\pi k'n/N} \quad (2.40)$$

$$\sum_{n=0}^{N-1} x[n] e^{-j2\pi k'n/N} = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} \hat{X}[k] e^{j2\pi(k-k')n/N} \quad (2.41)$$

$$\sum_{n=0}^{N-1} x[n] e^{-j2\pi k'n/N} = \frac{1}{N} \sum_{k=0}^{N-1} \hat{X}[k] N\delta_{k,k'} \quad (2.42)$$

$$\hat{X}[k'] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi k'n/N} \quad (2.43)$$

This is the inner product of our vector $x[n]$ with one of our orthogonal basis vectors! **It is important to remember that n and k are sample number and sample frequency respectively, and when working with actual data it is useful to convert to actual temporal/spatial values and frequencies.**

2.4 Continuous Fourier Transform

We are now going to discuss what is known as the Continuous Fourier Transform, commonly just Fourier Transform. This can be thought of as the limit of the Continuous Fourier Series as the period T is extended to ∞ . The FT is useful because it allows to analyze the frequency content of non-periodic functions! If we start with the Continuous Fourier Series expressions:

$$c_m = \frac{1}{T} \int_{-T/2}^{T/2} dt f(t) e^{-j2\pi mt/T} \quad \& \quad f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j2\pi nt/T} \quad (2.44)$$

We can move the $\frac{1}{T}$ to the temporal construction equation as follows:

$$c_m = \int_{-T/2}^{T/2} dt f(t) e^{-j2\pi mt/T} \quad \& \quad f(t) = \sum_{n=-\infty}^{\infty} \frac{c_n}{T} e^{j2\pi nt/T} \quad (2.45)$$

Then, taking the limit as $T \rightarrow \infty$ of the left equation:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} dt f(t) e^{-j2\pi \xi t} \quad (2.46)$$

Where we have defined ξ to be the (now continuous) frequency $\frac{m}{T}$. The temporal function reconstruction then becomes:

$$f(t) = \sum_{n=-\infty}^{\infty} \frac{1}{T} \hat{f}\left(\frac{n}{T}\right) e^{j2\pi nt/T} \quad (2.47)$$

When taking the limit of $T \rightarrow \infty$ this sum is interpreted as the Riemann sum of the argument with step size $\frac{1}{T}$

$$f(t) = \lim_{T \rightarrow \infty} \sum_{n=-\infty}^{\infty} \frac{1}{T} \hat{f}\left(\frac{n}{T}\right) e^{j2\pi nt/T} = \int_{-\infty}^{\infty} d\xi \hat{f}(\xi) e^{j2\pi \xi t} \quad (2.48)$$

It is common to use the Fourier Transform in terms of angular frequency $\omega = 2\pi\xi$, so we can do a quick change of variables to get the preferred form.

$$\boxed{\hat{f}(\omega) = \int_{-\infty}^{\infty} dt f(t) e^{-j\omega t}} \quad (2.49)$$

$$\boxed{f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \hat{f}(\omega) e^{j\omega t}} \quad (2.50)$$

The Fourier Transform $\hat{f}(\omega)$ of a function carries with it all of the information needed to describe that function. We are simply changing our basis from spatial/temporal to harmonic. The formula for constructing $\hat{f}(\omega)$ should also make sense intuitively because continuous complex exponentials form a basis for the infinite dimensional real number line! Therefore, we should be able to *pick out* the coefficients of this basis by taking the inner product with each of the basis vectors!

2.5 Discrete Time Fourier Transform

The Discrete Time Fourier Transform is considered the extension of DFT as $N \rightarrow \infty$. The $\frac{k}{N}$ becomes a continuous variable ξ , similar to how the Fourier Series was extended to the Fourier Transform. The DTFT is calculated as follows:

$$\lim_{N \rightarrow \infty} \hat{X}[k] = \lim_{N \rightarrow \infty} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \quad (2.51)$$

The following sum will extend from $-\infty$ to ∞ because the sum from 0 to N is actually a sum over any single period of our input signal.

$$\hat{X}(\xi) = \sum_{n=-\infty}^{\infty} x[n] e^{-j2\pi \xi n} \quad (2.52)$$

We again prefer to use angular frequency and define the new variable $\Omega = 2\pi\xi$.

$$\boxed{\hat{X}(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n}} \quad (2.53)$$

If $N \rightarrow \infty$, how should we calculate $x[n]$ from our DTFT? We will make the same extension to a Riemann sum as we did for the Fourier Transform.

$$x[n] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} \hat{X}[k] e^{j2\pi kn/N} \quad (2.54)$$

$$x[n] = \int_0^1 d\xi \hat{X}(\xi) e^{j2\pi \xi n} \quad (2.55)$$

And finally switch to $\Omega = 2\pi\xi$.

$$x[n] = \frac{1}{2\pi} \int_0^{2\pi} d\Omega \hat{X}(\Omega) e^{j\Omega n} \quad (2.56)$$

It is important to note that the DFT can be considered uniform samples of the DTFT, and that zero-padding a signal before taking the DFT brings it closer to the DTFT. A visual representation of this can be found [here](#)

3 Relationship between Convolution & Multiplication

One very important part of signal analysis is the fact that multiplication in the frequency domain corresponds to convolution in the spatial/temporal domain, and vice versa. This is an extremely important tool when it comes to finding the output and impulse response of LTI systems.

3.1 Aperiodic Continuous Functions

We can show that multiplication in the frequency domain is equal to convolution of continuous functions in the time/spatial domain as follows:

Suppose we multiply the Fourier transforms of two functions $f(t)$ and $g(t)$, what is the resulting temporal function?

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \hat{f}(\omega) \hat{g}(\omega) e^{j\omega t} = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \int_{-\infty}^{\infty} dt_1 f(t_1) e^{-j\omega t_1} \int_{-\infty}^{\infty} dt_2 g(t_2) e^{-j\omega t_2} e^{j\omega t} \quad (3.1)$$

$$= \int_{-\infty}^{\infty} dt_1 f(t_1) \int_{-\infty}^{\infty} dt_2 g(t_2) \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega e^{j\omega(t-t_1-t_2)} \quad (3.2)$$

The next step uses the Fourier transform of $\delta(t - t_0)$.

$$= \int_{-\infty}^{\infty} dt_1 f(t_1) \int_{-\infty}^{\infty} dt_2 g(t_2) \delta(t - t_1 - t_2) \quad (3.3)$$

$$= \int_{-\infty}^{\infty} dt_1 f(t_1) g(t - t_1) \quad (3.4)$$

Therefore,

$$\boxed{\hat{f}(\omega) \hat{g}(\omega) \Longleftrightarrow f(t) * g(t)} \quad (3.5)$$

The inverse relationship can also be shown (multiplication in the temporal domain corresponds to convolution in the frequency domain).

Why do we care? We can do many useful things with this tool, such as find the output of a continuous LTI system by multiplying the FT of the input and unit impulse response! We can also filter an incoming signal (attenuate the power of certain frequencies) by having a very specific impulse response! More on filtering later.

3.2 Aperiodic Discrete Functions

We can show that multiplication in the frequency domain is equal to convolution of discrete functions in the time/spatial domain as follows:

Suppose we multiply the DTFTs of two signals $f[n]$ and $g[n]$, what is the resulting signal?

$$\frac{1}{2\pi} \int_0^{2\pi} d\Omega \hat{F}(\Omega) \hat{G}(\Omega) e^{j\Omega n} = \frac{1}{2\pi} \int_0^{2\pi} d\Omega \sum_{n_1=-\infty}^{\infty} f[n_1] e^{-j\Omega n_1} \sum_{n_2=-\infty}^{\infty} g[n_2] e^{-j\Omega n_2} e^{j\Omega n} \quad (3.6)$$

$$= \sum_{n_1=-\infty}^{\infty} f[n_1] \sum_{n_2=-\infty}^{\infty} g[n_2] \frac{1}{2\pi} \int_0^{2\pi} d\Omega e^{j\Omega(n-n_1-n_2)} \quad (3.7)$$

The next step uses the DFT of $\delta[n - n_0]$.

$$= \sum_{n_1=-\infty}^{\infty} f[n_1] \sum_{n_2=-\infty}^{\infty} g[n_2] \delta[n - n_1 - n_2] \quad (3.8)$$

$$= \sum_{n_1=-\infty}^{\infty} f[n_1]g[n - n_1] \quad (3.9)$$

Therefore,

$$\boxed{\hat{F}(\Omega)\hat{G}(\Omega) \iff f[n] * g[n]} \quad (3.10)$$

Again, the inverse relationship is also true.

We can use this tool easily manipulate and solve our LTI systems!

3.3 Periodic Continuous Functions

This useful property is a little different when dealing with Periodic functions, as we will see below with the Fourier Series. What if we multiply each value of the Fourier Series of two (periodic in T) signals $f(t)$ and $g(t)$, with Fourier Series coefficients a_n and b_n respectively. We will call the resulting temporal function $s(t)$.

$$s(t) = \sum_{n=-\infty}^{\infty} a_n b_n e^{j2\pi n t/T} \quad (3.11)$$

$$s(t) = \sum_{n=-\infty}^{\infty} \frac{1}{T} \int_{-T/2}^{T/2} dt_1 f(t_1) e^{-j2\pi n t_1/T} \frac{1}{T} \int_{-T/2}^{T/2} dt_2 g(t_2) e^{-j2\pi n t_2/T} e^{j2\pi n t/T} \quad (3.12)$$

$$s(t) = \frac{1}{T} \int_{-T/2}^{T/2} dt_1 f(t_1) \int_{-T/2}^{T/2} dt_2 g(t_2) \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{j2\pi n(t-t_1-t_2)/T} \quad (3.13)$$

The next step uses the Fourier Series of a periodic $\delta(t - t_0)$.

$$s(t) = \frac{1}{T} \int_{-T/2}^{T/2} dt_1 f(t_1) \int_{-T/2}^{T/2} dt_2 g(t_2) \sum_{k=-\infty}^{\infty} \delta(t - t_1 - t_2 - kT) \quad (3.14)$$

$$s(t) = \frac{1}{T} \int_{-T/2}^{T/2} dt_1 f(t_1) \sum_{k=-\infty}^{\infty} g(t - t_1 - kT) \quad (3.15)$$

Note that

$$\sum_{k=-\infty}^{\infty} g(t - t_1 - kT) \quad (3.16)$$

is just periodic and shifted $g(t)$! Therefore, we can rewrite the eq. 107 as:

$$s(t) = \frac{1}{T} \int_{-T/2}^{T/2} dt_1 f(t_1) g((t - t_1) \bmod T) \quad (3.17)$$

$$\boxed{a_n b_n \iff \frac{1}{T} f(t) \circledast g(t)} \quad (3.18)$$

This operation is defined as periodic or **circular convolution**, and is denoted by the $\circledast$ operator. The inverse relationship is also true in this case as well, (multiplication in time corresponds to circular convolution in frequency).

3.4 Periodic Discrete Functions

The same property applies to Discrete Periodic Functions, or functions that are assumed to be periodic by taking the Discrete Fourier Transform. If we multiply each value of the DFT of two signals $f[n]$ and $g[n]$, and then take the inverse DFT, we will again be left with the circular convolution of the two signals. If we denote the resulting signal as $s[n]$, this can be proved similarly to how we proved the circular convolution property of the Fourier Series.

$$s[n] = \frac{1}{N} \sum_{k=0}^{N-1} \hat{F}[k] \hat{G}[k] e^{j2\pi kn/N} \quad (3.19)$$

$$s[n] = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{n_1=0}^{N-1} f[n_1] e^{-j2\pi kn_1/N} \sum_{n_2=0}^{N-1} g[n_2] e^{-j2\pi kn_2/N} e^{j2\pi kn/N} \quad (3.20)$$

$$s[n] = \sum_{n_1=0}^{N-1} f[n_1] \sum_{n_2=0}^{N-1} g[n_2] \frac{1}{N} \sum_{k=0}^{N-1} e^{j2\pi k(n-n_1-n_2)/N} \quad (3.21)$$

The next step uses the Discrete Fourier Transform of a periodic $\delta[t - t_0]$.

$$s[n] = \sum_{n_1=0}^{N-1} f[n_1] \sum_{n_2=0}^{N-1} g[n_2] \sum_{k=-\infty}^{\infty} \delta[n - n_1 - n_2 - kN] \quad (3.22)$$

$$s[n] = \sum_{n_1=0}^{N-1} f[n_1] \sum_{k=-\infty}^{\infty} g[n - n_1 - kN] \quad (3.23)$$

Note that

$$\sum_{k=-\infty}^{\infty} g[n - n_1 - kN] \quad (3.24)$$

is just periodic and shifted $g[n]$! Therefore, we can rewrite the eq. 115 as:

$$s[n] = \sum_{n_1=0}^{N-1} f[n_1] g[(n - n_1) \bmod N] \quad (3.25)$$

$$\boxed{\hat{F}[k] \hat{G}[k] \iff f[n] \circledast g[n]} \quad (3.26)$$

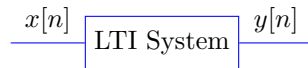
The inverse relationship is also true in this case as well, (multiplication in time corresponds to circular convolution in frequency).

3.5 Application to LTI Systems

We now have a new and extremely powerful tool. Suppose we know the impulse response of a particular LTI system, and we know the input to the system. We have already learned that we can calculate the output that system by convolving our input with our impulse response (section 1.4). Alternatively, we can now take the Fourier Transform of our input and impulse response, an $\mathcal{O}(n \log(n))$ operation for the DFT, multiply them, and simply take the inverse Fourier Transform to find out the output!

3.5.1 Discrete

For example, suppose we have an input to a discrete system $x[n]$, and impulse response $h[n]$ as shown below:



We can easily calculate $y[n]$ by performing the following steps. First, find the DTFT of the input $x[n]$ and the unit impulse response $h[n]$:

$$\hat{X}(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} \quad \& \quad \hat{H}(\Omega) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\Omega n} \quad (3.27)$$

Then simply take the inverse DTFT of their product to find $y[n]$!

$$y[n] = \frac{1}{2\pi} \int_0^{2\pi} d\Omega \hat{X}(\Omega) \hat{H}(\Omega) e^{j\Omega n} \quad (3.28)$$

You can also simply describe $y[n]$ by its frequency space representation $\hat{Y}(\Omega)$.

$$\hat{Y}(\Omega) = \hat{H}(\Omega) \hat{X}(\Omega) \quad (3.29)$$

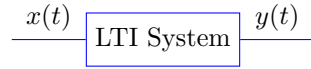
This is starting to look very similar to the operator notation described in chapter 1!

Example:

Suppose you are given a signal that represents an audio recording in from an area that has a very pronounced echo. You are also given the original audio signal without the echo. The relationship between the input audio signal and the output audio signal can be modeled as an LTI system! Therefore, if you solve for the unit impulse response of the system, you can theoretically remove the echo from any recorded signal!

3.5.2 Continuous

The same methodology applies to continuous LTI systems as well. For example, suppose we have an input to a continuous system $x(t)$, and impulse response $h(t)$ as shown below:



We can easily calculate $y(t)$ by performing the following steps. First, find the Fourier Transform of the input $x(t)$ and the unit impulse response $h(t)$:

$$\hat{x}(\omega) = \int_{-\infty}^{\infty} dt x(t) e^{-j\omega t} \quad \& \quad \hat{h}(\omega) = \int_{-\infty}^{\infty} dt h(t) e^{-j\omega t} \quad (3.30)$$

Then simply take the inverse Fourier Transform of their product to find $y(t)$!

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \hat{x}(\omega) \hat{h}(\omega) e^{j\omega t} \quad (3.31)$$

You can also simply describe $y(t)$ by its frequency space representation $\hat{y}(\omega)$.

$$\hat{y}(\omega) = \hat{h}(\omega) \hat{x}(\omega) \quad (3.32)$$

Many applications of this will be described when we discuss filtering.

3.6 More on Circular Convolution

The same LTI system application applies to the Discrete Fourier Transform & Fourier Series, unfortunately things are much more complicated due to the nature of circular convolution and the fact that out signals need to be the same length (each Fourier coefficient needs to be multiplied by another).

One way to apply the LTI system methodology is by zero padding your signals so that they are not only the same length, but twice as long as the longest signal! This is because the number of non-zero values

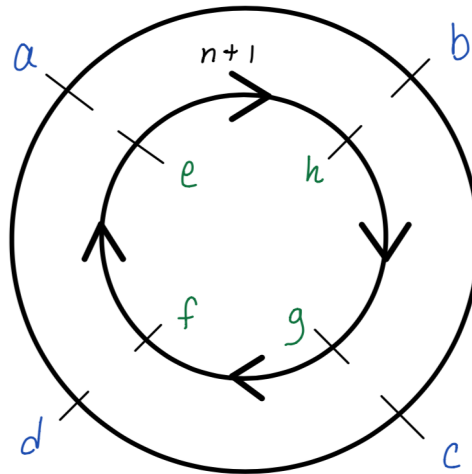
in the convolution of two signals of length N is $2N - 1$. The problem with circular convolution is shown graphically below.

Suppose we want to calculate the circular convolution of two signals of the same length $x[n]$ and $y[n]$:

$$x[n] = [a, b, c, d]$$

$$y[n] = [e, f, g, h]$$

One more intuitive way to think about this is by the following diagram:



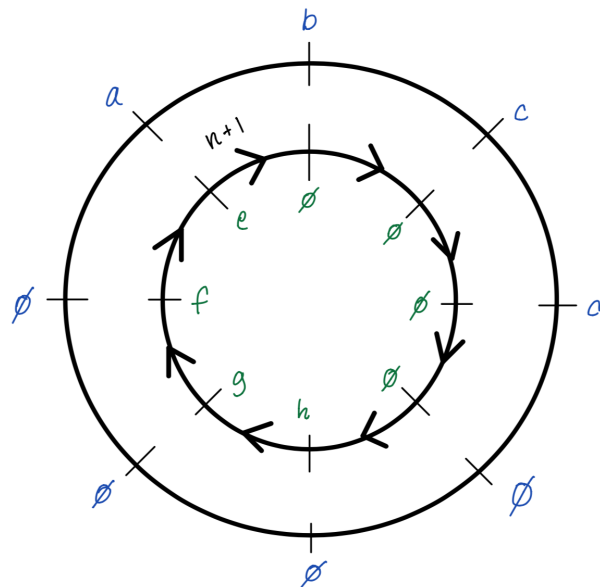
The signals $x[n]$ & $y[n]$ are represented in blue and green respectively. We can calculate the circular convolution of the two signals by multiplying all of the values that are adjacent along the two circles! We can advance to the next result of the convolution by rotating the inner circle clockwise as shown by the arrows. This diagram should clearly show you that there will be unwanted overlap during the convolution calculation.

As you can see, the result of such a convolution is most definitely different from the convolution of an input with an impulse response to find the output. If we simply zero pad each signal to twice their original length, we will end up with the proper (non-circular) convolution.

$$x[n] = [a, b, c, d, 0, 0, 0, 0]$$

$$y[n] = [e, f, g, h, 0, 0, 0, 0]$$

Resulting in the new diagram:



Now that the two signals have been zero-padded to twice their length, as the inner circle rotates around to $N = 8$ different positions, there will be no unwanted overlap, and the result of our circular convolution will be the same as the result of the output of an LTI system!

4 Simple Filtering & Bode Plots

An extremely useful application of the fact that convolution in time domain corresponds to multiplication in the frequency domain is the ability to remove certain frequency components from LTI inputs! We will call this process **filtering**.

Example:

Suppose we are making a high power audio system and we want to send the low frequency components to a different speaker than the high frequency components? Without the use of filtering we might need separate signals representing each part of the music, each recorded separately.

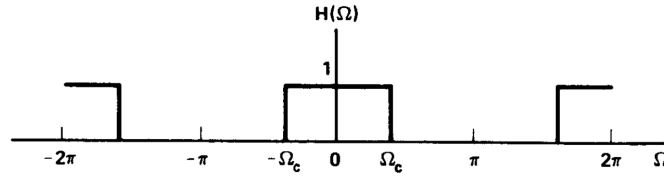
So how do we go about filtering signals?

4.1 Discrete Time

In order to accomplish filtering with discrete signals it makes most sense to use the Discrete Time Fourier Transform as opposed to the Discrete Fourier Transform. The DFT requires many conditions to be met in order for us to apply a filter; we must know the length of the input and the unit impulse response of our filter must be the same, we have to deal with the unfortunate effects of circular convolution, etc. We will only go into detail regarding simple filters designed using the DTFT.

Suppose we want to remove all of the frequency components with a magnitude greater than Ω_C of a discrete input signal. This operation corresponds to multiplying the DTFT of our input, $\hat{X}(\Omega)$, with the following DTFT:

$$\hat{H}(\Omega) \equiv \begin{cases} 1 & |\Omega| < \Omega_C \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$



We refer to this as a **lowpass filter**. The region with value 1 is referred to as the **bandpass region**, and the region with value zero is referred to as the **bandstop region**. In fact, this particular filter is an *ideal lowpass filter*, since it perfectly performs the lowpass operation with infinite cutoff slope and completely flat bandpass and bandstop regions.

If we want to perform this filtering operation on our incoming discrete signal $x[n]$, we can start with our convolution/multiplication relationship.

$$x[n] * h[n] \iff \hat{X}(\Omega) \hat{H}(\Omega) \quad (4.2)$$

Therefore, we must find the impulse response of our ideal lowpass filter by taking the Inverse DFT of $\hat{H}(\Omega)$:

$$h[n] = \frac{1}{2\pi} \int_{-\Omega_C}^{\Omega_C} d\Omega e^{j\Omega n} \quad (4.3)$$

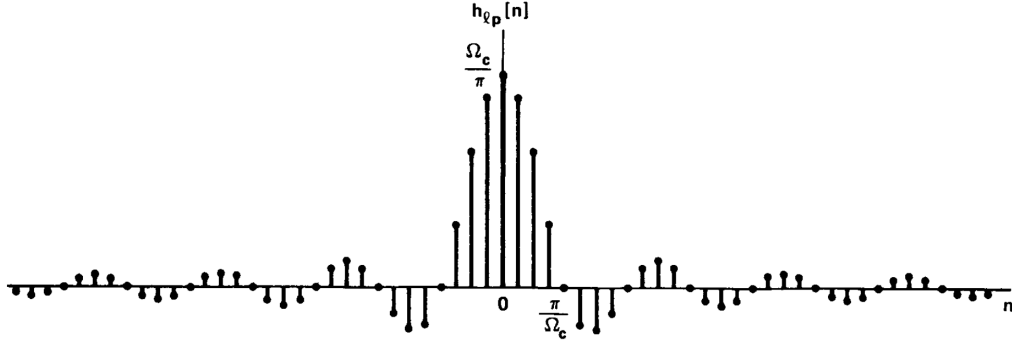
$$h[n] = \frac{1}{j2\pi n} [e^{j\Omega_C n} - e^{-j\Omega_C n}] \quad (4.4)$$

$$h[n] = \frac{1}{j2\pi n} [e^{j\Omega_C n} - e^{-j\Omega_C n}] \quad (4.5)$$

$$h[n] = \frac{2j \sin(\Omega_C n)}{j2\pi n} \quad (4.6)$$

$$h[n] = \frac{\sin(\Omega_C n)}{\pi n} \quad (4.7)$$

The impulse response $h[n]$ of our filter is known as a *sinc function* and is represented graphically below:



Now, when this impulse response is convolved with any input signal $x[n]$, it will completely remove all frequency components with a magnitude greater than Ω_C .

*It is important to note that this impulse response is non-causal, meaning the output at time n depends on inputs at times ahead of n . In fact, an ideal lowpass filter depends on inputs at times infinitely ahead of n . This can not be good, and is one of the reasons why an ideal lowpass filter is not used in practice. Another reason why the ideal lowpass filter would not be used is because its impulse response is infinite in the other direction; the convolution would take forever to implement.

We will not go into further depth regarding discrete filtering as the design of advanced filters is a very difficult and complex topic.

You will calculate the impulse response of an ideal highpass filter in a homework assignment.

4.2 Continuous Time

Filters are also a very important concept in continuous time systems. We again start with our convolution/multiplication relationship.

$$f(t) * g(t) \iff \hat{f}(\omega)\hat{g}(\omega) \quad (4.8)$$

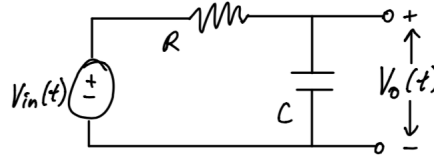
Clearly, the frequency space of the input is scaled by the frequency space of the unit impulse response of our system before reaching the output. As stated before, when applied appropriately, this operation is called filtering.

Suppose we know the differential equation governing a continuous time system, that differential equation tells us the relationship between the input and the output of our system. The differential equation is the temporal or spatial representation of the continuous LTI system.

We will analyze a continuous time lowpass filter. An ideal lowpass filter in continuous time is much more difficult to conceptualize, so we will analyze filtering applications of the relationship between convolution and multiplication using AC circuits!

Example: Passive RC lowpass filter.

Suppose we are given the following RC circuit, with the following governing differential equation.



$$\dot{V}_0(t) + \frac{V_0(t)}{RC} = \frac{V_{in}(t)}{RC} \quad (4.9)$$

If this equation governs the time domain relationship between input and output, then its Fourier Transform should govern the frequency domain relationship!

$$j\omega\hat{V}_0(\omega) + \frac{\hat{V}_0(\omega)}{RC} = \frac{\hat{V}_{in}(\omega)}{RC} \quad (4.10)$$

Note that temporal derivative is represented in the frequency domain by multiplication by $j\omega$, you will prove this in a homework assignment. What we learned was that in the frequency domain the an LTI system is defined by $\hat{Y}(\omega) = \hat{H}(\omega)\hat{X}(\omega)$. Therefore, equation 4.9 can be manipulated to that form!

$$(j\omega RC + 1)\hat{V}_0(\omega) = \hat{V}_{in}(\omega) \quad (4.11)$$

$$\hat{V}_0(\omega) = \hat{V}_{in}(\omega) \frac{1}{(j\omega RC + 1)} \quad (4.12)$$

Therefore

$$\hat{H}(\omega) = \frac{1}{(j\omega RC + 1)} \quad (4.13)$$

We have found the Fourier Transform of the impulse response of the RC circuit! Why is this useful? If this $\hat{H}(\omega)$ is multiplied by $\hat{V}_{in}(\omega)$, it scales the frequency space of our input! Now in order to make physical sense of this multiplication it is necessary to manipulate the complex expression for $\hat{H}(\omega)$ into polar form:

$$\hat{H}(\omega) = \frac{1}{(j\omega RC + 1)} = \frac{1}{\sqrt{1 + (\omega RC)^2}} e^{-j \tan^{-1}(\omega RC)} \quad (4.14)$$

We can now see that the Fourier Transform of the input is multiplied by a magnitude and angle:

$$|\hat{H}(\omega)| = \frac{1}{\sqrt{1 + (\omega RC)^2}} \quad \& \quad \angle \hat{H}(\omega) = -\tan^{-1}(\omega RC) \quad (4.15)$$

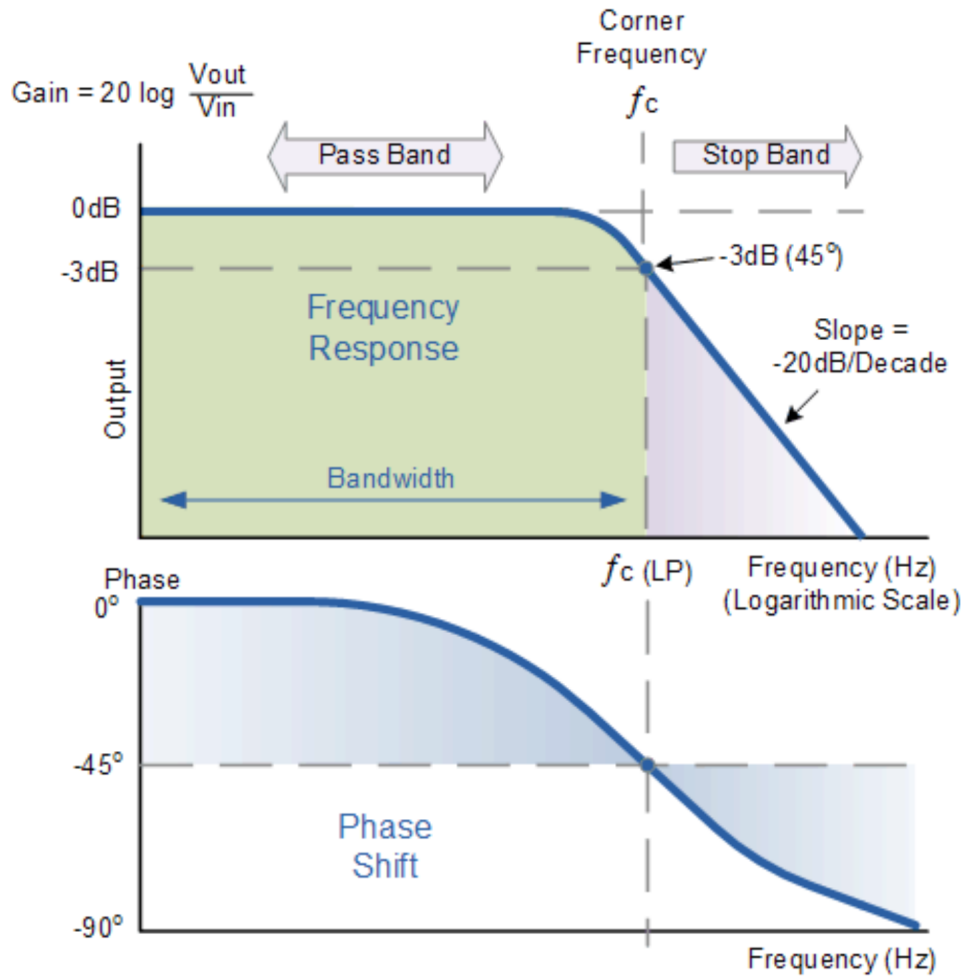
This RC Circuit is called a continuous lowpass filter because as ω increases from 0 to ∞ , the magnitude of $\hat{H}(\omega)$ goes from 1 to 0. Since the magnitude of $\hat{H}(\omega)$ scales the Fourier Transform of the input, the input frequency space is *filtered* by our RC circuit before reaching the output! We can use these types of circuits to manipulate the frequency content of continuous signals.

You will calculate the frequency response of a similar mechanical system in a homework assignment.

4.3 Bode Plots

Bode plots represent 10 times the base 10 logarithm of the power of the frequency response of filters on a log log scale. If you have some filter with frequency space representation $\hat{H}(\omega)$, then the bode plot of such a filter has $\log(\omega)$ on the x-axis and $20 \log |\hat{H}(\omega)|$ on the y-axis. The multiplication by 20 instead of 10 is due to power being proportional to the square of voltage or current, and properties of the logarithm.

For example, we can show the bode plot of the lowpass RC filter from the last section as follows:



The plot of the $|\hat{H}(\omega)|$ is represented on the top and the plot of $\angle \hat{H}(\omega)$ is represented on the bottom. Note the phase plot is not on a log log scale. You can now clearly see why this particular circuit is a lowpass filter, (when multiplied by the Fourier Transform of the input, high frequency components are greatly attenuated).

The cutoff frequency f_c is defined as the frequency at which the filter attenuates the power of an input signal by $\frac{1}{2}$. The rolloff rate is also important for filters and is defined as the slope of the stop band section of the bode plot on a log log scale.

Bode plots are an extremely common way of representing the frequency response of filters!

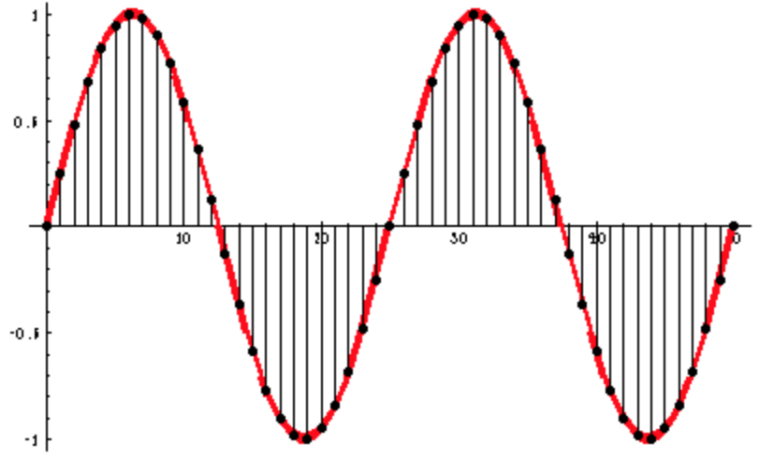
5 Ideal Sampling

Digital hardware operates in discrete steps, but the real world is analog, so we must have a way of converting continuous time signals into digital signals that computers can use. This process is known as sampling and we can model it mathematically as follows:

Suppose we have a continuous time signal $y(t)$ that is sampled every T seconds, we can represent that sampled signal as $y(nT)$ for integer valued n . The process is also represented by the following diagram.



A visual representation of this process is included below:



The samples of the red sinusoid are represented as the black lines. Although sampling seems like a seemingly painless process, we will explore the unapparent complexity and problems with the process.

5.1 Aliasing

Suppose we have a sinusoidal continuous signal of frequency f that we would like to sample every T seconds.

$$y(t) \rightarrow y(nT) \quad (5.1)$$

$$\sin(2\pi ft) \rightarrow \sin(2\pi fnT) \quad (5.2)$$

Now what if we would like to sample a signal at a frequency $f + M/T$ every T seconds, where M is an integer.

$$y(t) \rightarrow y(nT) \quad (5.3)$$

$$\sin(2\pi(f + \frac{M}{T})t) \rightarrow \sin(2\pi(f + \frac{M}{T})nT) \quad (5.4)$$

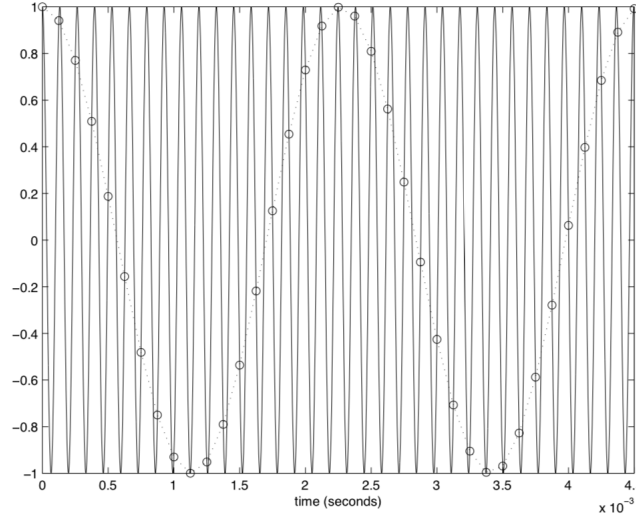
$$= \sin(2\pi fnT + 2\pi \frac{M}{T} nT) \quad (5.5)$$

$$= \sin(2\pi fnT + 2\pi nM) \quad (5.6)$$

$$= \sin(2\pi fnT) \quad (5.7)$$

As you can see from equations 5.2 and 5.7, the two sampled signals look exactly the same! Therefore, we can ascertain that any signal with any two signals with a difference in frequency of M/T will look exactly the same to our sampler! This phenomenon is called **aliasing**, and is a huge problem when it comes to sampling continuous signals from the physical world.

A visual representation of this phenomenon is included below:



The continuous high frequency sinusoid is sampled at every small circle, and the small circles trace out a very different frequency signal than our continuous time signal!

5.2 Analog to Digital

We can use the Fourier Transform machinery we've developed to analyze the process of sampling much more formally.

The process of sampling a continuous time signal can be mathematically represented by multiplying the input signal by a periodic pulse train (periodic train of delta functions). For example, given an input signal $x(t)$, sample period T , and sampled signal $y(t)$:

$$y(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (5.8)$$

$$y(t) = \sum_{n=-\infty}^{\infty} x(nT) \delta(t - nT) \quad (5.9)$$

We see that our sampled signal is represented by a sequence of weighted delta functions. This abstraction allows for the construction of individually valued points from our continuously valued input.

We have learned that convolution in time corresponds to multiplication in frequency, *and* that multiplication in time corresponds to convolution in frequency (although this was not proven).

$$f(t)g(t) \iff \frac{1}{2\pi} \hat{f}(\omega) * \hat{g}(\omega) \quad (5.10)$$

So how can we represent the frequency space, $\hat{f}(\omega)$, of our sampled signal $y(t)$? First, we must find the Fourier Transform of a pulse train.

It is important to note that the easiest way to find the Fourier Transform of a period function is through its Fourier Series representation. The Fourier Transform of a pulse train periodic in T is calculated as follows:

We first find the Fourier Series coefficients a_n .

$$a_n = \frac{1}{T} \int_{-T/2}^{T/2} dt \delta(t) e^{-j2\pi nt/T} = \frac{1}{T} \quad (5.11)$$

*An interactive plot of the reconstruction of the periodic delta function can be found [here](#).

We need to find the Fourier transform that gives the same result as the Fourier series. Now since the inverse fourier **series** is of the following form:

$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi kt/T} \quad (5.12)$$

And the inverse fourier **transform** is of the following form:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \hat{f}(\omega) e^{j\omega t} \quad (5.13)$$

$\hat{f}(\omega)$ must take the following form in order to get the same $f(t)$:

$$\hat{f}(\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi k}{T}\right) \quad (5.14)$$

Working through the math to show the desired result:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \hat{f}(\omega) e^{j\omega t} \quad (5.15)$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi k}{T}\right) e^{j\omega t} \quad (5.16)$$

$$f(t) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} d\omega \delta\left(\omega - \frac{2\pi k}{T}\right) e^{j\omega t} \quad (5.17)$$

$$f(t) = \frac{1}{T} \sum_{k=-\infty}^{\infty} e^{j2\pi kt/T} \quad (5.18)$$

Where we can clearly see that c_n is equal to $1/T$ as expected.

Now that we know the Fourier transform of our sampling signal (the periodic pulse train), we can find the Fourier transform of our sampled signal $\hat{y}(\omega)$ by convolution:

$$\hat{y}(\omega) = \frac{1}{2\pi} \hat{x}(\omega) * \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi k}{T}\right) \quad (5.19)$$

$$\hat{y}(\omega) = \frac{1}{T} \int_{-\infty}^{\infty} d\omega' \sum_{k=-\infty}^{\infty} \delta\left(\omega' - \frac{2\pi k}{T}\right) \hat{x}(\omega - \omega') \quad (5.20)$$

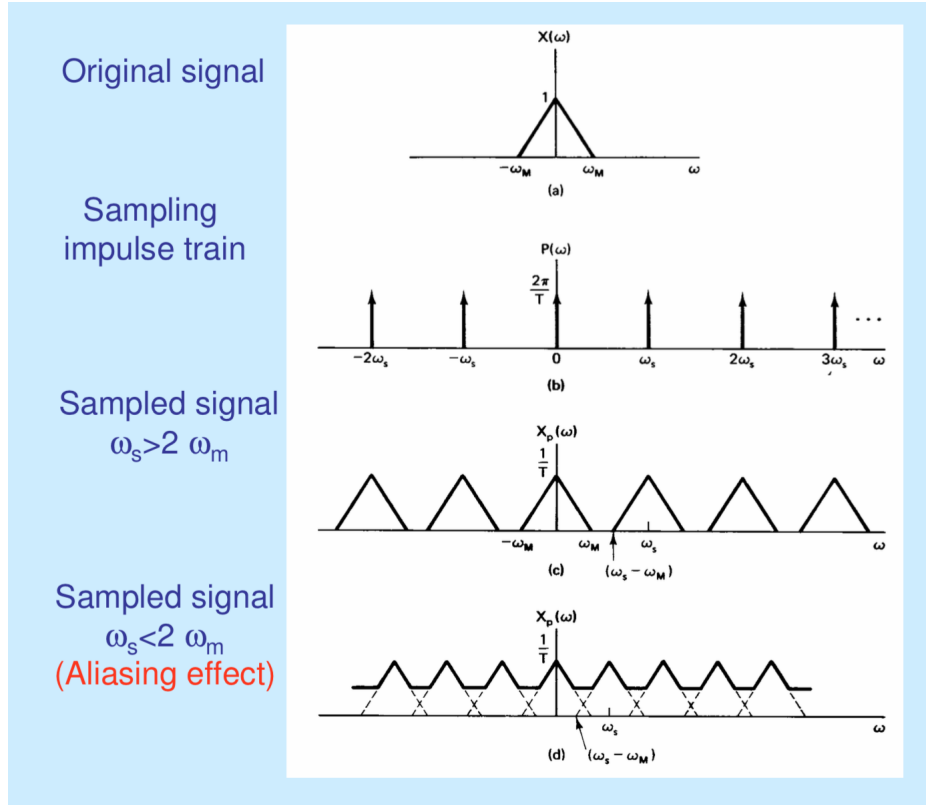
$$\hat{y}(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} d\omega' \delta\left(\omega' - \frac{2\pi k}{T}\right) \hat{x}(\omega - \omega') \quad (5.21)$$

$$\hat{y}(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \hat{x}\left(\omega - \frac{2\pi k}{T}\right) \quad (5.22)$$

If we define our sampling angular frequency as $\omega_s = 2\pi/T$, the the equation becomes:

$$\hat{y}(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \hat{x}(\omega - \omega_s k) \quad (5.23)$$

We can see that the Fourier transform of our sampled continuous function is equal to a sum of scaled and shifted copies of our original input signal's Fourier transform, and the period of $\hat{y}(\omega)$ is our sampling frequency ω_s ! The process is represented visually below:



*The overlap in the in the fourth plot is the negative effect produced by the aliasing discussed in section 5.1. This will be analyzed further soon.

Now we have only discussed the continuous time representation of our sampling system, but we are trying to construct discrete signals. We said that sampling is of a continuous time signal was represented by the following:

$$y(t) \rightarrow x(nT) \quad (5.24)$$

Remember that x is just sampled y , therefore $x(nT)$ is basically a discrete signal $x[n]$.

$$y(t) \rightarrow y(nT) \rightarrow x(nT) \rightarrow x[n] \quad (5.25)$$

This process is abstracted away into a ADC (Analog to Digital Converter).



Note that we have already shown that the DTFT is periodic every $\Omega = 2\pi$. So it makes sense that the CFT of our sampled signal is periodic in exactly the same thing (the units of Ω are radians per sample, and the definition of ω_s is angular sampling frequency)! Also note that **the units of Ω is radians/sample, and the units of ω_s is radians/second.** *In conclusion, the CFT of our sampled signal is the DTFT of that discrete signal (up to a scale factor).***

Now if we are analyzing the frequency content of the newly constructed discrete signal, based on the above image, it is clear that any frequency content of our original signal that overlaps with the sequential copies is no longer the same frequency content in the DTFT. Therefore, our sampling process will map higher frequency content to lower frequencies which are not actually present. This consequence is the reason why **we must not have any frequency content in our original signal greater than half of our sampling frequency!** The minimum sampling rate in order to capture all of the frequency content of a signal accurately is known as the **Nyquist Rate**, and the maximum frequency that can be accurately described by a sampled signal is known as the **Nyquist Frequency**,

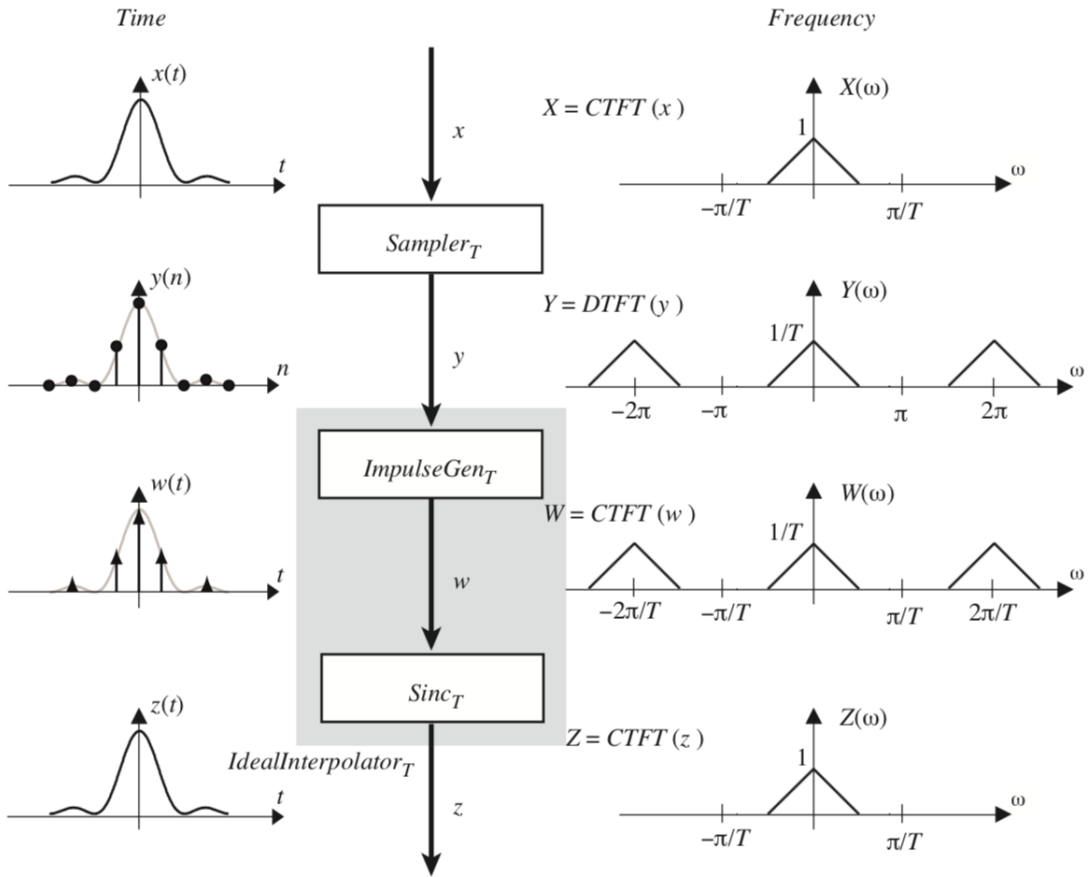
5.3 Digital to Analog

Now what if we want to convert our digital signal back into the original continuous signal? We must first convert our digital $x[n]$ back into a continuous function. This process can be abstracted away into some ADC (Analog to Digital Converter). **Think of this as recreating the scaled pulse train produced at the beginning of the sampling process.**



We know the frequency space representation of our input, and the frequency space representation of our sampled signal, so why not apply a filter to our sampled signal that will recover our original signal!

We can apply an ideal bandpass filter to our sampled signal and reconstruct our original signal as shown below:



The band pass filter must have the following equation:

$$h(\omega) \equiv \begin{cases} T & |\omega| < \omega_s/2 = \pi/T \\ 0 & \text{else} \end{cases} \quad (5.26)$$

The temporal representation of this ideal bandpass filter can be calculated to be a continuous sinc function scaled by T .

$$h(t) = \frac{T}{2\pi} \int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} d\omega e^{j\omega t} \quad (5.27)$$

$$h(t) = \frac{T}{j2\pi n} [e^{j\omega t}]_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} \quad (5.28)$$

$$h(t) = \frac{T}{j2\pi t} (e^{j\frac{\omega_s}{2}t} - e^{-j\frac{\omega_s}{2}t}) \quad (5.29)$$

$$h(t) = \frac{T2j \sin(\frac{\omega_s}{2}t)}{j2\pi t} \quad (5.30)$$

$$h(t) = \frac{T \sin(\frac{\omega_s}{2}t)}{\pi t} \quad (5.31)$$

We have shown how to perfectly reconstruct our sampled signal.

*Note, to apply this filter in the time domain we must convolve our sampled signal with $h(t)$, a function with infinite range! This is one of the reasons why, in practice, ideal reconstruction is difficult to implement and other methods are used (interpolation, zero-order hold, etc). You can read about interpolation [here](#), and zero-order hold [here](#).

6 Laplace & Z-Transforms, Feedback Control

6.1 Introduction

We will now analyze generalizations of the Fourier Transform and Discrete Time Fourier Transform. These new tools are extremely useful for system analysis and predicting system stability. It is important to note that, throughout this entire section, we are assuming causal systems (the Z/Laplace transform completely describes the function of time without knowing the region of convergence).

6.1.1 Unilateral Laplace Transform

The Laplace transform is another way of representing continuous functions. We can think of it as taking the $j\omega$ from the exponent of the Fourier Transform and generalizing it to a new value $s = \sigma + j\omega$ with both real and imaginary values. We define the Laplace transform as follows:

$$F(s) = \int_{0^-}^{\infty} dt f(t) e^{-st} \quad (6.1)$$

The integral only starts at $t = 0^-$ to because we are now focusing of system analysis, in which are our signals can be made to start at $t = 0$. **Notice that since $s = \sigma + j\omega$ we can find the Laplace transform for functions which we could not find the Fourier transform (unbounded at ∞), as long as the exponential e^{-st} decays faster than $f(t)$ grows!**

It can be shown using the techniques of Complex Analysis that the inverse Laplace Transform is the following:

$$f(t) = \frac{1}{2\pi j} \lim_{T \rightarrow \infty} \int_{\gamma-jT}^{\gamma+jT} ds F(s) e^{st} \quad (6.2)$$

A Couple Useful Properties:

time-shift (prove as HW):

$$F(t - t_0) = e^{-st_0} F(t) \quad (6.3)$$

temporal derivatives:

$$F(s) = \int_{0^-}^{\infty} dt f^{(n)}(t) e^{-st} \quad (6.4)$$

Use integration by parts multiple times:

$$F(s) = f^{(n-1)}(t) e^{-st} \Big|_{0^-}^{\infty} + s \int_{0^-}^{\infty} dt f^{(n-1)}(t) e^{-st} \quad (6.5)$$

$$F(s) = f^{(n-1)}(\infty) e^{-s\infty} - f^{(n-1)}(0^-) + s \int_{0^-}^{\infty} dt f^{(n-1)}(t) e^{-st} \quad (6.6)$$

Where the $t = \infty$ term must equal zero for the Laplace transform to exist.

$$F(s) = -f^{(n-1)}(0^-) + s \int_{0^-}^{\infty} dt f^{(n-1)}(t) e^{-st} \quad (6.7)$$

In general, we can express the time derivative in the Laplace domain as follows:

$$\mathcal{L}\{f'\} = sF(s) - f(0^-) \quad (6.8)$$

$$\mathcal{L}\{f''\} = s^2 F(s) - sf(0^-) - f'(0^-) \quad (6.9)$$

$$\mathcal{L}\{f^{(n)}\} = s^n F(s) - s^{n-1} f(0^-) - s^{n-2} f'(0^-) - \dots - f^{(n-1)}(0^-) \quad (6.10)$$

Where $\mathcal{L}(f)$ denotes the Laplace transform.

Examples:

What is the Laplace transform of e^{at} ?

$$F(s) = \int_{0-}^{\infty} dt e^{at} e^{-st} \quad (6.11)$$

$$F(s) = \int_{0-}^{\infty} dt e^{(a-s)t} \quad (6.12)$$

Note that this integral will only converge if $\Re(a-s)$ is less than 0.

$$F(s) = \frac{1}{a-s} [e^{(a-s)t}]_{0-}^{\infty} \quad (6.13)$$

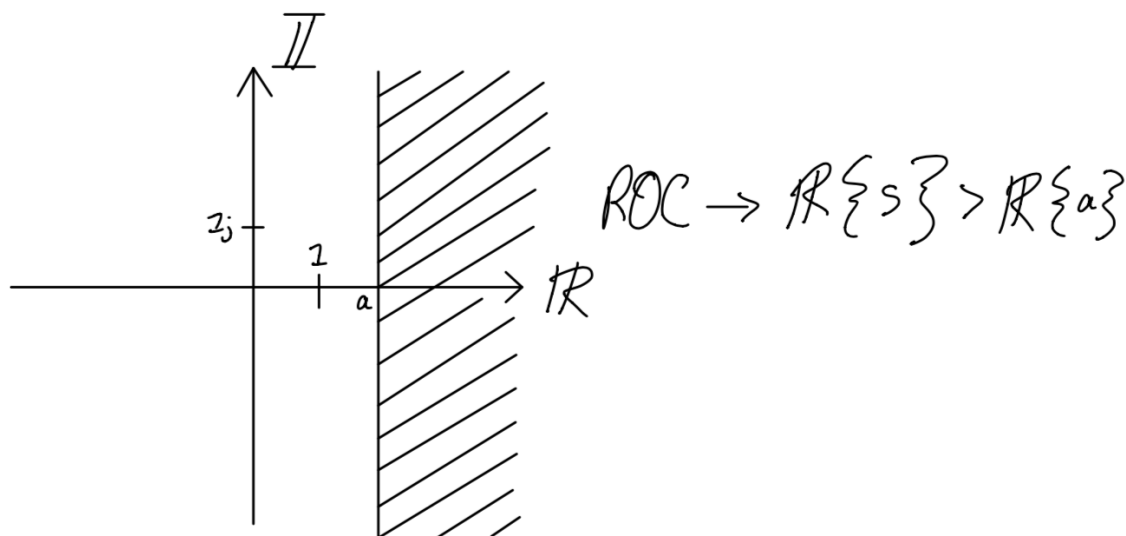
$$F(s) = \frac{1}{s-a} \quad (6.14)$$

Find the Laplace transform of te^{at} as HW. How would you find the Laplace transform of $\sin(at)$ or $\cos(at)$?

Region of Convergence:

In the previous examples the Laplace transform of e^{at} only converged if $\Re(a-s)$ was less than 0. We will call the range of s for which the Laplace transform is non-infinite the **region of convergence**.

Taking $\Re(a)$ to be a positive number, the region of convergence of the Laplace transform of e^{at} is the following:

**6.1.2 Bilateral Z Transform**

The Z transform is another way of representing Discrete signals (which will result in the familiar operator notation from section 1). We define the Z transform as follows:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \quad (6.15)$$

Where z is a complex number. This is similar to the DTFT if we think of z as a replacement for $e^{-j\Omega}$.

It can be shown using the techniques of Complex Analysis that the inverse Z Transform is the following:

$$f(t) = \frac{1}{2\pi j} \oint_C dz X(z) z^{n-1} \quad (6.16)$$

Useful Properties:

n-shift (prove as HW):

$$\mathcal{Z}\{x[n - n_0]\} = z^{-n_0} X(z) \quad (6.17)$$

Where $\mathcal{Z}(x)$ denotes the Z transform.

Examples:

What is the Z transform of $x[n] = a^n u[n]$?

$$X(z) = \sum_{n=-\infty}^{\infty} a^n u[n] z^{-n} \quad (6.18)$$

$$X(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} \left(\frac{a}{z}\right)^n \quad (6.19)$$

Note that this sum will only converge if $|\frac{a}{z}|$ is less than 1.

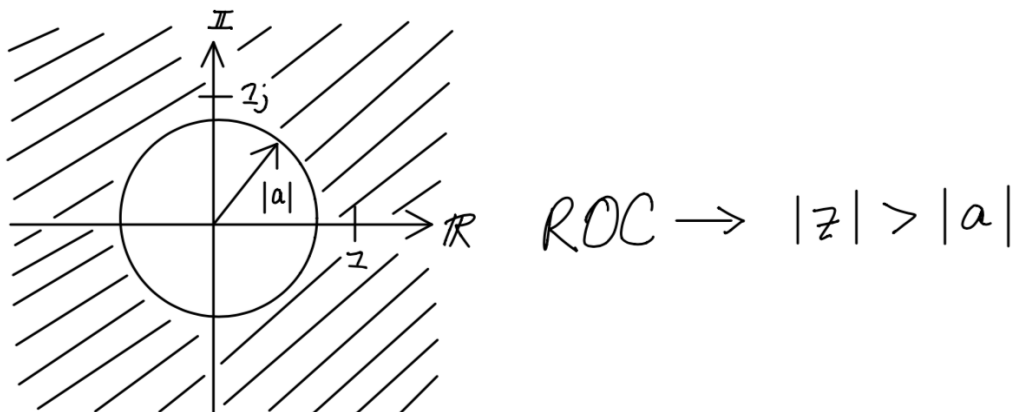
$$X(z) = \frac{1}{1 - a/z} = \frac{1}{1 - az^{-1}} = \frac{z}{z - a} \quad (6.20)$$

Does this look like the discrete time transfer function of any familiar LTI system?

Region of Convergence:

The values for which the Z transform of a discrete time signal converge are enclosed in the **region of convergence** of the Z transform.

Taking $|a|$ to be less than 1, the region of convergence of the Z transform of a^n is the following:



6.2 Convolution & Multiplication

6.2.1 Continuous

It can be shown that convolution in the temporal domain corresponds to multiplication in the frequency domain of the Laplace transform as follows.

$$h(t) = f(t) * g(t) \quad (6.21)$$

$$h(t) = \int_{0-}^{\infty} dt_1 f(t_1)g(t - t_1) \quad (6.22)$$

Taking the Laplace transform of both sides (using the shift property)

$$H(s) = \int_{0-}^{\infty} dt_1 f(t_1)e^{-t_1s}G(s) \quad (6.23)$$

$$H(s) = F(s)G(s) \quad (6.24)$$

6.2.2 Discrete

It can be shown that convolution in the temporal domain corresponds to multiplication in the frequency domain of the Z transform as follows.

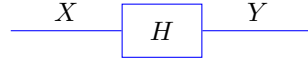
$$h[n] = f[n] * g[n] = \sum_{k=-\infty}^{\infty} f[k]g[n - k] \quad (6.25)$$

We now take the Z transform of both sides using the superposition and delay property:

$$H(z) = \sum_{k=-\infty}^{\infty} f[k]z^{-k}G(z) \quad (6.26)$$

$$H(z) = F(z)G(z) \quad (6.27)$$

Now if we think back to discrete LTI systems, this is exactly what we were doing with the operator notation! We simply rename variables as follows and we can immediately recognize X as our input signal, H as our system functional, and Y as our output!



$$Y(z) = H(z)X(z) \quad (6.28)$$

We simply didn't explicitly calculate $Y(z)$ and $X(z)$ because they are dependent on unknown input.

6.3 Application to LTI Systems

Now that we've been introduced to the Laplace and Z transforms, a couple of their useful properties, and the fact their relationship with convolution/multiplication a clear connection to LTI systems should be apparent. We've already discussed using the Fourier transform to find solutions to LTI systems, but we now have an even more generalized and powerful tool that can be directly related to our algebraic expressions for discrete time systems as well as handle unbounded functions.

6.3.1 Discrete Time Systems

If you look back to section 1.1, it should be clear to you that taking the Z transform of a discrete LTI system (starting with the functional representation) corresponds to writing the LTI system in what we called "operator" notation. A brief reminder of this process:

$$y[n] = ax[n] - by[n - 1] \quad (6.29)$$

Taking the Z transform of both sides (using time shift property and linearity):

$$Y(z) = aX(z) - bz^{-1}Y(z) \quad (6.30)$$

We have learned that we can find the output of LTI systems using the convolution of the input with the unit sample response, we have also learned that convolution in the sample domain corresponds to

multiplication in the Z domain, therefore the product of $X(z)$ with some function $H(z)$ must equal to $Y(z)$.

$$Y(z) + bz^{-1}Y(z) = aX(z) \quad (6.31)$$

$$(1 + bz^{-1})Y(z) = aX(z) \quad (6.32)$$

$$\frac{Y(z)}{X(z)} = H(z) = \frac{a}{1 + bz^{-1}} \quad (6.33)$$

Importantly, $H(z)$ must equal the Z transform of the unit sample response of this system! We can use the Z transform to solve for the unit sample response of any LTI system using the following technique.

Finding Impulse Response:

1: Find the system functional representation of any LTI system given in general form (where for brevity we will denote the Z transform of a signal $f[n]$ as F):

$$H = \frac{Y}{X} = \frac{\sum_{j=0}^n b_j z^{-j}}{\sum_{i=0}^m a_i z^{-i}} \quad (6.34)$$

2: By the fundamental theorem of algebra, any order n polynomial has n complex roots (counted with multiplicity). With this in mind, we can rewrite any transfer function in the following form:

$$H = \frac{Y}{X} = \frac{\prod_{j=0}^n (1 - c_j z^{-1})}{\prod_{i=0}^m (1 - d_i z^{-1})} \quad (6.35)$$

*The values of z for which the numerator is 0, (c_j), are known as the **zeros** of the transfer function, and the values of z for which the denominator is 0, (d_i), are known as the **poles** of the transfer function.

3: Decompose the denominator of the previous expression into a sum using partial fraction decomposition.

$$H = \frac{Y}{X} = \prod_{j=0}^n (1 - c_j z^{-1}) \sum_{i=0}^m \frac{A_i}{1 - d_i z^{-1}} \quad (6.36)$$

4: In general, this sum of fractions can be inverted to the time domain representation using a variety of techniques (z transform table, analytical), but for our simple functions the following method will suffice (although it will **NOT** always work, due to complexities with repeated roots and non-exponential functions):

Note that a z transform of the form:

$$H_i(z) = \frac{a_i z^{-n_0}}{1 - d_i z^{-1}} \quad (6.37)$$

corresponds to the following discrete function (since the time shift and linearity tell us how to handle the numerator). The z^{-n_0} delays the function by n_0 and the a_i is simply a scale factor.

$$h_i[n] = a_i d_i^{n-n_0} \quad (6.38)$$

So following partial fraction decomposition, and taking advantage of the linearity of the Z transform, we can clearly see that after multiplying out the numerator of expression 6.37, the impulse response $h[n]$ can be represented as a sum of scaled and shifted exponentials (that may be complex).

*I have also attached a table of inverse Z transforms to make your life easier (although light partial fraction decomposition may still be required in the homework).

STABILITY:

Note that now we have a well established method of solving for the impulse response of a discrete LTI system, we should be able to easily see whether or not the system is stable (does not tend towards $\pm\infty$,

or oscillate forever). We have seen that each pole of the transfer function (d_i) corresponds to the base of an exponential function (or a more complex function which we have not discussed) in the impulse response, so it should be clear that the impulse response will only be stable if the magnitude of all poles (d_i) is less than 1.

For example: If the system we just solved above in equation 6.37 and 6.38, is the following:

$$H_i(z) = \frac{a_i z^{-n_0}}{1 - d_i z^{-1}} \iff h_i[n] = a_i d_i^{n-n_0} \quad (6.39)$$

We can clearly see that $h_i[n]$ will grow to infinity or oscillate forever unless $|d_i|$ is less than 1.

NOTE: The region of convergence of the Z transform also tells us something useful about stability. Since the response of a system can be characterized by a sum of scaled and shifted exponential functions (as shown above), they will each contribute to the region of convergence, (actually the exponential with the largest magnitude base will govern the ROC. **Therefore, if the region of convergence contains the unit circle, the system is stable.** Convince yourself this is true.

6.3.2 Continuous Time Systems

We can also analyze continuous time systems in the Laplace domain. The time-shift, temporal derivative, and convolution/multiplication properties we've demonstrated have extremely useful implications when it comes to solving differential equations and finding the impulse response of continuous LTI systems.

We can start with the usual representation of a continuous time system as a ordinary linear time invariant differential equation:

$$y'(t) + cy(t) = dx(t) \quad (6.40)$$

Taking the Laplace transform of both sides utilizing the time derivative rule explained above yields the following:

$$sY(s) - y(0^-) + cY(s) = dX(s) \quad (6.41)$$

Forcing rest initial conditions allows us to manipulate the equation into the following form:

$$Y(s) = \frac{d}{s+c} X(s) \quad (6.42)$$

Again, we've shown that multiplication in the Laplace domain corresponds to convolution in the frequency domain, so $X(s)$ must be multiplied by the Laplace transform $H(s)$ of the impulse response $h[n]$ of our continuous LTI system!

$$Y(s) = \frac{d}{s+c} X(s) = H(s)X(s) \quad (6.43)$$

$$H(s) = \frac{d}{s+c} \quad (6.44)$$

Finding Impulse Response

After taking the Laplace Transform of an ordinary LTI differential equation, given rest initial conditions, the quotient of output and the input is defined as the **continuous transfer function**.

$$\frac{Y(s)}{X(s)} = H(s) \quad (6.45)$$

*Note: evaluating this continuous transfer function at $s = j\omega$ exposes the *filtering* properties of the continuous system!

We can find the impulse response of an ordinary LTI differential equation in a very similar method to that of discrete LTI systems (assume rest initial conditions):

Given an ordinary LTI differential equation of the form:

$$\cdots a_2 y''(t) + a_1 y'(t) + a_0 y(t) = b_0 x(t) + b_1 x'(t) + b_2 x''(t) \cdots \quad (6.46)$$

1: Find the transfer function of any ordinary LTI differential equation in general form (where for brevity we will denote the Laplace transform of a function $f(t)$ as $F(s)$):

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\sum_{j=0}^n b_j s^j}{\sum_{i=0}^m a_i s^i} \quad (6.47)$$

2: By the fundamental theorem of algebra, any order n polynomial has n complex roots (counted with multiplicity). With this in mind, we can rewrite any transfer function in the following form:

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\prod_{j=0}^n (1 - c_j s)}{\prod_{i=0}^m (1 - d_i s)} \quad (6.48)$$

*The values of s for which the numerator is 0, $(c_j)^{-1}$, are known as the **zeros** of the transfer function, and the values of s for which the denominator is 0, $(d_i)^{-1}$, are known as the **poles** of the transfer function.

3: Decompose the denominator of the previous expression into a sum using partial fraction decomposition.

$$H(s) = \frac{Y(s)}{X(s)} = \prod_{j=0}^n (1 - c_j s) \sum_{i=0}^m \frac{A_i}{1 - d_i s} \quad (6.49)$$

4: In general, this sum of fractions can be inverted to the time domain representation using a variety of techniques (Laplace transform table, analytical), but for our simple functions the following method will suffice (although it will **NOT** always work, due to complexities with repeated roots and non-exponential functions):

Note that a Laplace transform of the form:

$$H(s) = \frac{s^{n_0} a_i}{d_i - s} \quad (6.50)$$

corresponds to the following continuous function

$$h(t) = a_i d_i^{n_0} e^{d_i t} \quad (6.51)$$

So following partial fraction decomposition, and taking advantage of the linearity of the Laplace transform, we can find the impulse response by summing the contributions of multiple fractions.

**Take advantage of the provided Laplace table when solving homework problems (although light partial fraction decomposition may still be required in the homework).

Solving Ordinary LTI Differential Equations

It is possible to completely solve many ordinary linear differential equations using the Laplace transform (they don't need to be time invariant) using the convolution/multiplication properties. You simply follow the same method as outlined for finding impulse response, but solve for $Y(s)$ instead of $Y(s)/X(s)$, (where Y is output and X is input).

example:

Solve $\dot{x} + 3x = e^{-t}$, $x(0^-) = 4$.

Start by taking the Laplace transform of both sides:

$$sX(s) - x(0^-) + 3X(s) = \frac{1}{1+s} \quad (6.52)$$

Plugging in initial conditions and solving for $X(s)$:

$$sX(s) + 3X(s) = \frac{1}{1+s} + 4 \quad (6.53)$$

$$X(s)(s+3) = \frac{1}{1+s} + 4 \quad (6.54)$$

$$X(s) = \frac{1}{(1+s)(s+3)} + \frac{4}{s+3} \quad (6.55)$$

We now need to perform partial fraction decomposition on the first term, yielding:

$$X(s) = \frac{1/2}{s+1} - \frac{1/2}{s+3} + \frac{4}{s+3} \quad (6.56)$$

$$X(s) = \frac{1/2}{s+1} + \frac{7/2}{s+3} \quad (6.57)$$

And finally looking up inverse Laplace relationship:

$$x(t) = \frac{e^{-t}}{2} + \frac{7e^{-3t}}{2} \quad (6.58)$$

Which satisfies the given differential equation and initial conditions.

7 PID Feedback Control

PID (Proportional, Integral, Derivative) Control is an effective way of improving the transient response of a system (settling time and overshoot) while also ensuring that there is minimal steady state error. In this section we will explore various techniques for controller design.

7.1 Root Locus

The root locus of a closed loop transfer function depicts how the poles of that system change as the gain (K) changes. We can manipulate the root locus by adding controllers to our system to move the closed-loop poles to a desired location.

Drawing the Root Locus:

To draw the root locus one must first obtain the transfer function of the open loop system. For this example we will use a simple system, with open loop transfer function

$$G(s) = \frac{(s + 4)}{(s + 1)(s + 2)} \quad (7.1)$$

and a proportional controller with variable gain K.

There are a few rules when drawing the root locus which allow us to easily do it by hand. They are:

Rule 1: If all poles and zeros of the open loop transfer function are on the real axis, a point s on the real axis is on the root locus if and only if there is an odd number of poles/zeros of G(s) to its right.

Rule 2: For K close to 0, the poles of the closed loop transfer function start at the poles of G(s) (the open loop transfer function).

Rule 3: As K approaches infinity, s on the root locus approaches the zeroes of G(s) and a finite number of asymptotes at angles

$$\theta_l = \frac{(2l + 1)\pi}{(n - m)}, l = 0, 1, \dots, (n - m - 1) \quad (7.2)$$

where n is the number of poles and m is the number of zeroes in the open loops transfer function. These asymptotes meet on the real axis at the centroid, which can be found using

$$\sigma = \frac{(p_1 + p_2 + \dots + p_n) - (z_1 + z_2 + \dots + z_m)}{(n - m)} \quad (7.3)$$

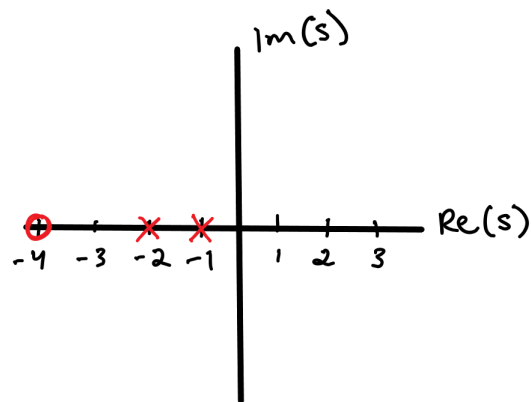
Rule 4: Between any two poles (or zeros) of G(s) on the real axis that are next to each other, there is an ODD number of breakaway points (points where two or more branches of the root locus come together, aka a repeated root, and then they separate). When they separate, they depart the real axis at an angle of 90 degrees. The breakaway points (s*) can be found using

$$\left. \frac{d}{ds} G(s) \right|_{(s=s^*)} = 0 \quad (7.4)$$

Now, back to our example system. For now, we will just look at the open loop transfer function to place our poles and zeroes:

$$G(s) = \frac{(s + 4)}{(s + 1)(s + 2)} \quad (7.5)$$

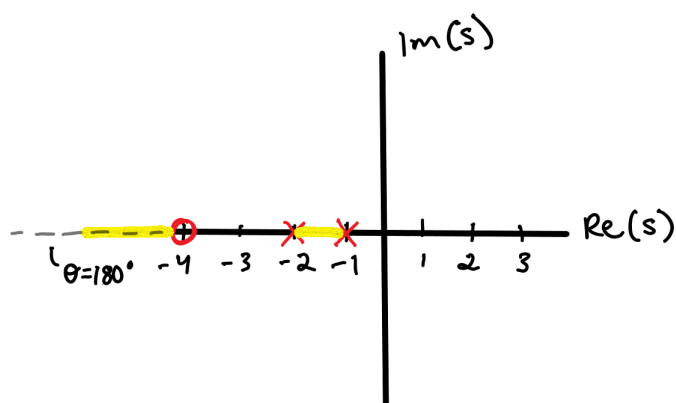
Our equation has two poles and one zero. (Zeroes are found using the numerator of the fraction, and poles using the denominator.) The poles are at -1 and -4, and the zero is at -2. Since there are no complex poles or zeroes, they all lie on the real axis. Poles are drawn as an x and zeros are drawn as an o.



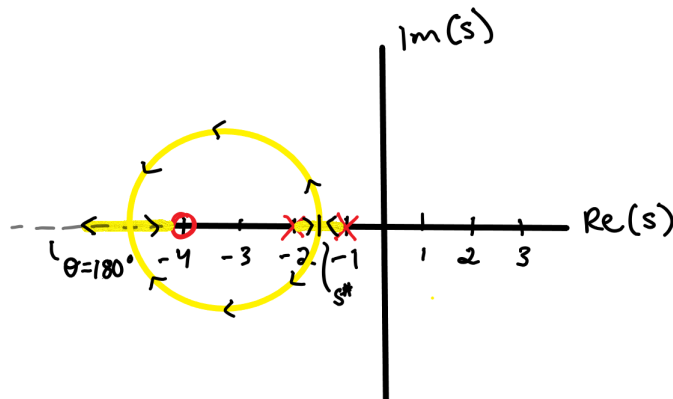
Next, let's find our asymptotes. In our case, $n=2$ and $m=1$, so we will only have one asymptote since $(n-m) = 1$. This asymptote exists at

$$\theta = \frac{(2l - 1)\pi}{n - m}$$

and is simply π (180°). Next we'll draw our root locus on the real axis. There is an odd number of poles and zeroes to the right between $(-2, -1)$ and $(-\infty, -4)$.



Finally, we know that there exists a breakaway point between the two poles of the function. We find this breakaway point by setting the derivative of $G(s)$ equal to zero and solving for s . Using this method, we find that the breakaway point between $(-2, -1)$ is -1.55 . Thus, we know that as the gain K of the controller increases the poles of the system will depart the real axis at -1.55 , but what will they do after they depart? We know that as K approaches infinity the poles of the closed loop system will approach our asymptote and the zero of the system. Therefore, the poles will depart the real axis at -1.55 and reconverge at some point past -4 , where they will split apart as one travels to $-\infty$ and the other approaches -4 .



If you wanted to know the exact gain value when a pole is in a certain place, you could use the magnitude condition, which states that:

$$|G(s)| = \frac{1}{K} \quad (7.6)$$

The magnitude condition is useful for when you have system requirements that give you a desired pole location and you wish to find the value of the gain K that will bring the closed-loop poles to that location.

7.2 Routh-Hurwitz Criterion

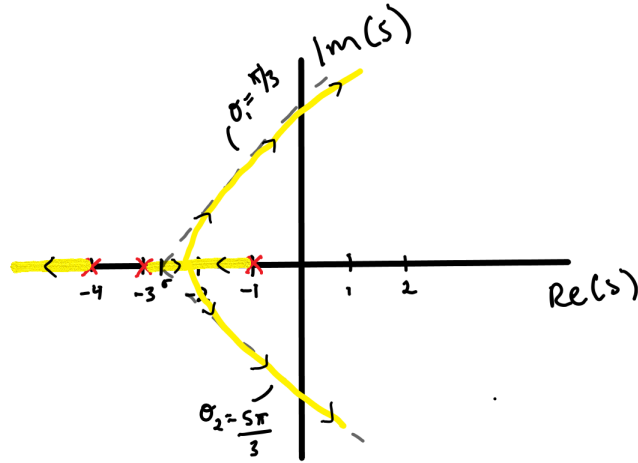
The Routh-Hurwitz criterion and table are a simple way for finding when the poles of a transfer function cross into the right-half plane and the system becomes unstable.

In order to use the Routh-Hurwitz criterion, you must find $g(s)$, which is the polynomial in the denominator of your closed loop transfer function. For example, say we had a plant with transfer function

$$P(s) = \frac{1}{(s+1)(s+3)(s+4)} \quad (7.7)$$

and a simple proportional controller with gain K .

If we draw our root locus, we can see that the system will clearly become unstable at some value of $K \neq 0$. We could use the magnitude condition to find this value, but it typically takes a while to compute the derivative of $G(s)$, and we would need to know at which point on the imaginary axis the root locus crosses, so we will use the Routh-Hurwitz criterion instead.



To find our $g(s)$, we write out our closed loop transfer function and find that it is:

$$g(s) = (s + 1)(s + 3)(s + 4) + K \quad (7.8)$$

which expands to

$$g(s) = s^3 + 8s^2 + 19s + (12 + K). \quad (7.9)$$

Next, we use the coefficients of $g(s)$ to construct our Routh-Hurwitz table. It is constructed as such:

$$\begin{array}{c|cccc} s^n & a_n & a_{n-2} & a_{n-4} & \dots \\ s^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & \dots \\ \cdot & b_1 & b_2 & b_3 & \dots \\ \cdot & c_1 & c_2 & c_3 & \dots \\ \cdot & \dots & \dots & \dots & \dots \end{array}$$

The a_i coefficients are simply the coefficients corresponding to the s^i values in $g(s)$. You can find the b and c coefficients using the following equations:

$$\begin{aligned} b_1 &\equiv \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}} & b_2 &\equiv \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}} \\ c_1 &\equiv \frac{b_1 a_{n-3} - a_{n-1} b_2}{b_1} & c_2 &\equiv \frac{b_1 a_{n-5} - a_{n-1} b_3}{b_1} \end{aligned}$$

In our case, the Routh Hurwitz table looks like this:

$$\begin{array}{r|rrr}
 s^3 & 1 & 19 & 0 \\
 s^2 & 8 & (12+K) & 0 \\
 s^1 & \frac{152 - (12+K)}{8} & 0 & 0 \\
 s^0 & (12+K) & 0 & 0
 \end{array}$$

Now, for what the RH table will tell us. For each sign change in the first column of coefficients (as you travel down the column), there is a pole of the closed loop transfer function on the right half plane (meaning the system is unstable). In our case, all the coefficients are positive so long as $0 < K < 140$. (This was found by setting the leading two coefficients that included a K greater than zero. For our purposes, we will only consider K greater than zero, so the last coefficient will be positive for all K values.)

7.3 PID Controller Design

PID Controllers are very useful for creating a desired response to any type of input. A good PID controller can minimize the overshoot of a system, while decreasing the settling time, while also minimizing steady-state error (and in some cases, eliminating it entirely). There is a relatively simple way to design PID controllers that we will cover in this section.

The Derivative ("D") part of the controller is what improves the transient response of the system, while the Integral "I" part of the controller reduces the steady state error.

The Procedure:

First, we will determine our desired pole location based on design specs. You can use these equations for settling time and percent overshoot to find your desired pole location given those parameters.

$$T_s = \frac{4}{Re(s)} \quad (7.10)$$

$$percentOS = e^{\frac{-(Re(s))\pi}{Im(s)}} * 100 \quad (7.11)$$

Once we have our desired pole location, we can begin designing our controller. We may also be told that we need a specific requirement for the steady state error, which will tell us how many poles we need at the origin. In this example, we will want no steady state error to a step input, which means we need one

pole at the origin.

Next, we draw our original root locus with a simple controller K. We also plot our desired pole locations. This will help us determine where we need to place the poles and zeros of our controller to move the root locus to our desired location.

We then incorporate our pole at the origin (to eliminate steady state error) into the plant itself. We then redraw our root locus to include this pole.

With this new root locus, we can calculate the deficiency angle φ . The deficiency angle will tell us where to place the zero in our controller. To calculate the deficiency angle, we simply draw lines from each existing pole and zero to our desired pole location, and measure the angle these lines create with the horizontal. The deficiency angle is calculated using the equation:

$$\varphi = -180^\circ - (-\theta_1 - \theta_2 - \dots - \theta_n) + (-\phi_1 - \phi_2 - \dots - \phi_m) \quad (7.12)$$

Where θ are the angles created by the poles and ϕ are the angles created by the zeros.

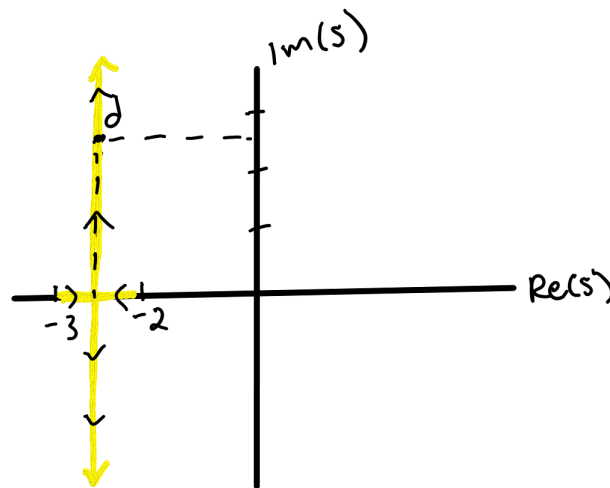
Finally, we place the zero of our controller on the real axis so that it creates the deficiency angle φ with the desired pole location. If the deficiency angle is greater than 60° , we can implement a double PD controller, in the form $C(s) = K(s+z)^2$, with each zero at $-z$ contributing $\frac{\varphi}{2}$. When the deficiency angle is too large, we run the risk of zero-pole cancellation altering our system. This ensures that the root locus will travel through our desired pole location.

EXAMPLE:

In this example, we are given design specs that our desired pole location is $d = -2.5 \pm 2.5i$, and we want our system to have zero steady state error to a step input (meaning we need one pole at the origin). Our plant transfer function is

$$P(s) = \frac{4}{(s+2)(s+3)} \quad (7.13)$$

First, we'll draw our root locus for a simple proportional controller with gain K.

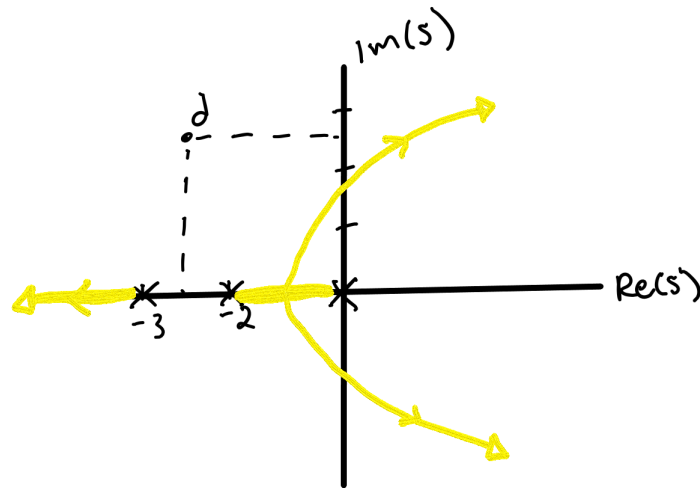


As you can see, the root locus does in fact cross through our desired pole location, but since there is no pole at the origin, the steady state error is not zero. To fix this, we must add a pole at the origin. Next,

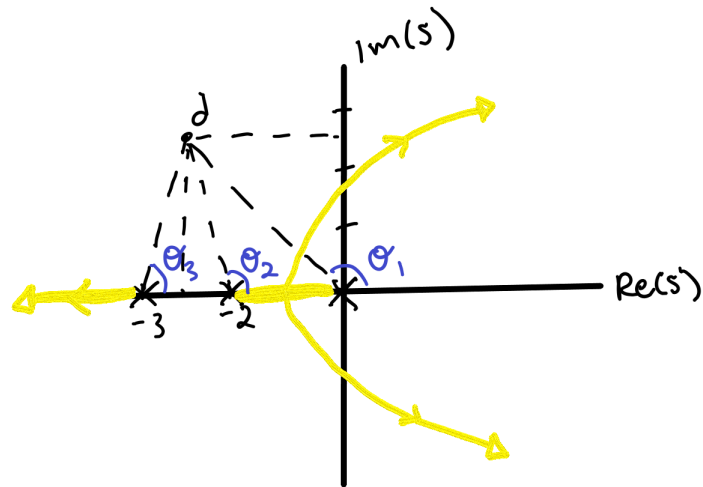
we incorporate our pole at the origin to the plant itself. Our new plant transfer function is:

$$P(s) = \frac{4}{s(s+2)(s+3)} \quad (7.14)$$

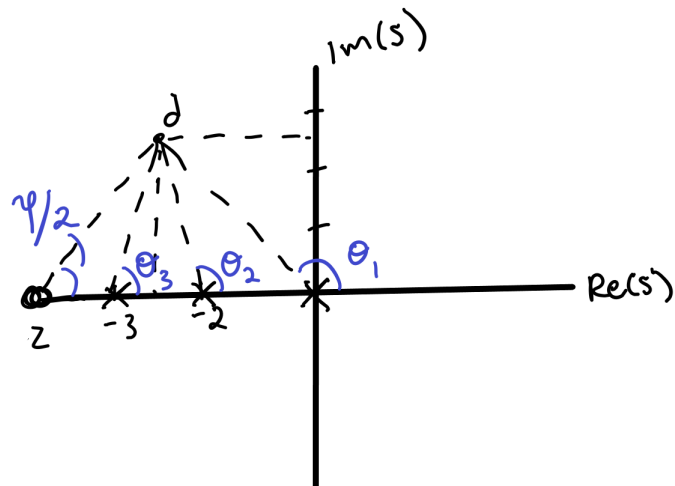
and we will draw our new root locus with this added pole:



Now, our system will have no steady state error, but the root locus no longer goes through the desired pole location. To fix this, we will design a controller of the form $C(s) = K(s+z)$ to "bend" the branches of our root locus through d. We calculate our deficiency angle by finding $\theta_1, \theta_2, \theta_3$ in relation to d.



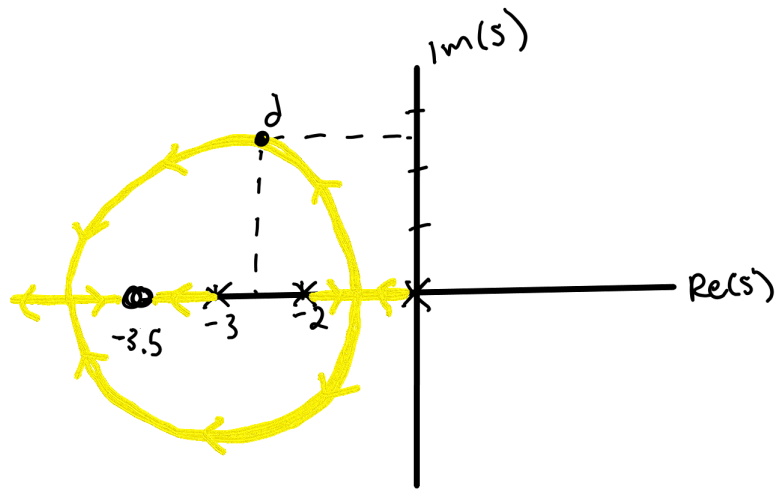
In our case, $\theta_1 = 135^\circ$ and $\theta_2 + \theta_3 = 180^\circ$. Therefore, our deficiency angle $\varphi = -180^\circ + 180^\circ + 135^\circ = 135^\circ$. Since our deficiency angle is $> 60^\circ$, we will divide it in half and use a double PD controller. $\frac{\varphi}{2} = 67.5^\circ$.



Using geometry, we find the location of z is:

$$\frac{2.5}{x} = \tan^{-1}(67.5^\circ), x = 1, z = -3.5 \quad (7.15)$$

We plot our new root locus, with the added double PD controller, and see that it does indeed fulfill our requirements:



The last thing we must do is find the value of K that will bring our poles to their desired location. We find this using the magnitude condition.

$$\left|G(s)\right| = \frac{1}{K}, s = d = -2.5 \pm 2.5i \quad (7.16)$$

We now know that our desired gain value is 0.79, and our final controller design is

$$C(s) = \frac{0.79(s + 3.5)^2}{s} \quad (7.17)$$

which is a PID controller!

8 Basic Signal Modulation

Signal modulation is a very useful method of transmitting information over mediums which they might not traverse well in their base form. AM (amplitude modulation) and FM (frequency modulation) radio are great examples of modulating signals to transmit audio signals over great distances. We will explore the basic characteristics of a few simple modulation schemes, and leave the detailed implementation and more complex schemes to higher level courses.

8.1 Amplitude Modulation

Amplitude modulation, as the name states, is the method of modulating the amplitude of some high frequency *carrier wave* with the information to be transmitted. As we well know, the information to be transmitted can be deconstructed into a sum of weighted harmonic waves with a certain bandwidth (width of frequency space representation). We will explore amplitude modulation of the following form:

Suppose the information we want to transmit is some signal $f(t)$, with Fourier transform $\hat{F}(\omega)$ with a restricted bandwidth B and center frequency ω_0 (note, real signals will exhibit frequencies around $-\omega_0$ as well):

$$f(t) \iff \hat{F}(\omega) \neq 0 \text{ for } ||\omega| - |\omega_0|| \leq B \quad (8.1)$$

We will call the much higher frequency wave we will modulate the amplitude of $c(t)$, and define it as a pure harmonic with frequency $\omega_c \gg \omega_0$:

$$c(t) = \cos(\omega_c t) \iff \hat{C}(\omega) = \pi(\delta(-\omega_c) + \delta(\omega_c)) \quad (8.2)$$

Defining $m(t)$ as the modulated signal, we need to modulate our carrier wave amplitude with the original signal in the following way:

$$m(t) = (A + 2Mf(t)) \cos(\omega_c t) \quad (8.3)$$

Where A and M are both real numbers defining amplitudes of parts of the modulated signal. This is the simple temporal representation of an amplitude modulated signal.

If we take the center frequency of the signal in the place of $f(t)$, we can see by trigonometric identities what such a signal might look like:

$$m(t) = (A + 2M \cos(\omega_0 t)) \cos(\omega_c t) \quad (8.4)$$

$$m(t) = A \cos(\omega_c t) + M(\cos((\omega_c + \omega_0)t) + \cos((\omega_c - \omega_0)t)) \quad (8.5)$$

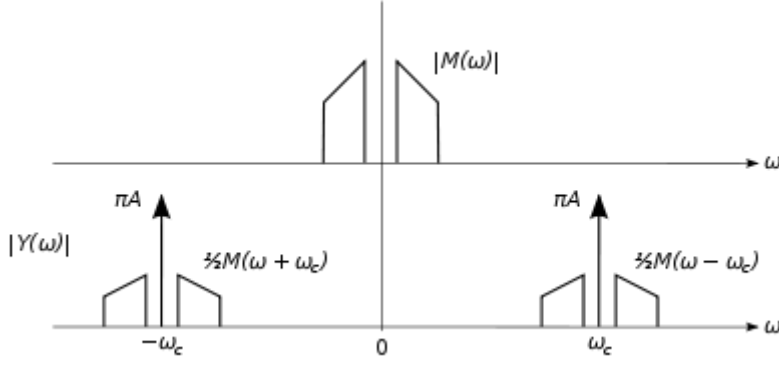
It is clear that we end up with 3 pure harmonics; one at ω_c , one at $\omega_c + \omega_0$, and one at $\omega_c - \omega_0$. An interactive video of what such a signal (evolving in time) would look like can be found [here](#).

Frequency Space:

How could we represent the modulated signal in the frequency domain, given that the carrier wave is multiplied with the original signal in the temporal domain?

We have learned that multiplication in time is the same thing as convolution in frequency, so we simply need to convolve the Fourier transforms of the carrier wave and of the modulated wave.

A visual representation of this process is shown below:



For this sake of this picture I found on the internet, the frequency space representations of the original signal and modulated signal are $M(\omega)$ and $Y(\omega)$ respectively. The Fourier transform of the original signal is convolved with the two delta functions that define a pure harmonic at frequency ω_c

Demodulation:

Mathematically, we can demodulate an AM signal in a very simple and intuitive way, (in practice the circuits that perform this action are not so simple).

Suppose we multiply our one band of our modulated signal by a pure harmonic of the same frequency and phase as the carrier wave (after filtering out the other parts):

$$\cos(\omega_c t) m_b(t) = \cos(\omega_c t) \cos((\omega_c - \omega_0)t) \quad (8.6)$$

Using the same trigonometric identity as before:

$$\cos(\omega_c t) m_b(t) \propto \cos(2\omega_c t) + \cos(\omega_0 t) \quad (8.7)$$

As you can see the resultant signal has frequency components at $2\omega_c$ (twice the carrier) and ω_0 (the original signal)! We simply need to filter out the high frequency content using a low pass filter.

8.2 Frequency Modulation

Frequency modulation is another technique used to transfer information on a carrier signal, except this time we modulate the *frequency* of the carrier signal instead of the amplitude.

Before discussing the frequency modulation, we need to define the **instantaneous frequency** of a harmonic wave. Usually, when we think about the frequency of a wave, we think about the following:

$$y(t) = \cos(2\pi f t) \quad (8.8)$$

Where the frequency of this harmonic wave is f . Taking the derivative of our signal *pulls out* the frequency:

$$\dot{y}(t) = -2\pi f \sin(2\pi f t) \quad (8.9)$$

So what if we have a signal of the following form:

$$y(t) = \cos\left(2\pi \int_0^t d\tau f(\tau)\right) \quad (8.10)$$

And take the derivative to *pull out* the frequency:

$$\dot{y} = -2\pi f(t) \sin\left(2\pi \int_0^t d\tau f(\tau)\right) \quad (8.11)$$

By this understanding, the frequency is $f(t)$, and is referred to as the **instantaneous frequency** of the harmonic.

With this method of thinking, we are now ready to understand frequency modulation as changing the instantaneous frequency of a carrier wave! Suppose the carrier wave was intrinsic frequency f_c , and we would like to modulate it with a signal $s(t)$.

$$f_c = \text{carrier frequency} \quad \& \quad s(t) = \text{information to be transmitted} \quad (8.12)$$

If we want to modulate the frequency of our carrier, we should define the instantaneous frequency as follows:

$$f(t) = f_c + f_\Delta s(t) \quad (8.13)$$

Now putting it all in a harmonic wave we get the following modulated signal:

$$y(t) = \cos \left(2\pi \int_0^t d\tau (f_c + f_\Delta s(\tau)) \right) \quad (8.14)$$

Where the instantaneous frequency changes with $s(t)$! The value constant value f_Δ is referred to as the *frequency deviation*, and determines the sensitivity of our carrier frequency to our information signal.

So what does this look like for a simple sine wave information signal with frequency f_m ?

$$s(t) = A_m \cos(2\pi f_m t) \quad (8.15)$$

$$\int_0^t d\tau s(\tau) = A_m \frac{\sin(2\pi f_m t)}{2\pi f_m} \quad (8.16)$$

After evaluating the modulation equation our final signal will be of the form:

$$y(t) = \cos \left(2\pi f_c t + \frac{f_\Delta A_m}{f_m} \sin(2\pi f_m t) \right) \quad (8.17)$$

To get a better sense of what this signal would actually look like, see [here](#).

8.3 Phase Modulation

Phase modulation is very similar to frequency modulation, except we simply don't integrate our information signal, (therefore modifying the phase of the carrier wave instead of the frequency).

Just like in AM and FM, we will have a carrier harmonic with angular frequency ω_c , of which we will modulate the phase our signal $s(t)$:

$$y(t) = \cos(\omega_c t + A_m s(t)) \quad (8.18)$$

If we take our information to be another harmonic of frequency ω_i , the modulated signal looks very familiar to FM:

$$y(t) = \cos(\omega_c t + A_m \cos(\omega_i t)) \quad (8.19)$$

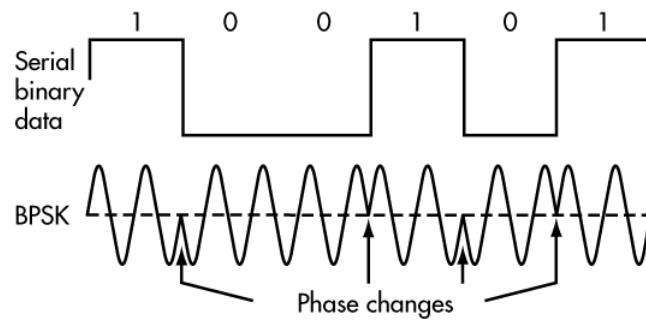
A great visual representation of a phase modulated signal can be found [here](#)

8.3.1 Binary Phase Shift Keying

One of the more simple means of transferring digital data using phase modulation is known as Binary Phase Shift Keying (BPSK). If the information we wish to transmit is simply a sequence of bits, then the function describing our modulated signal becomes much simpler:

$$y(t) = \cos(\omega_c t + \pi(n - 1)) \quad \text{where } n \in \{0, 1\} \quad (8.20)$$

Where n represents the value of the bit sequence we wish to transfer! A visual depiction of what such a signal would look like is shown below:



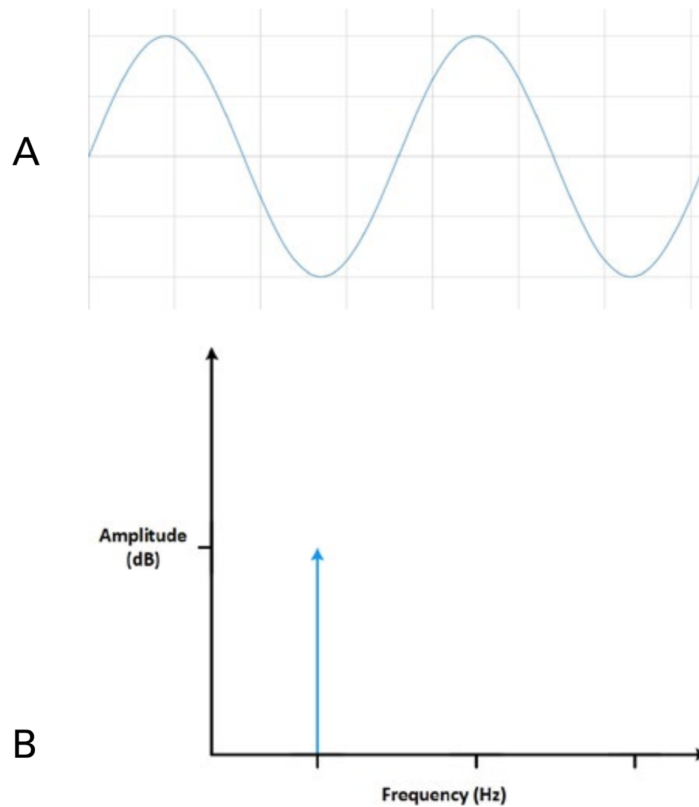
9 Other Applications

9.1 Spectrum Smearing & Windowing

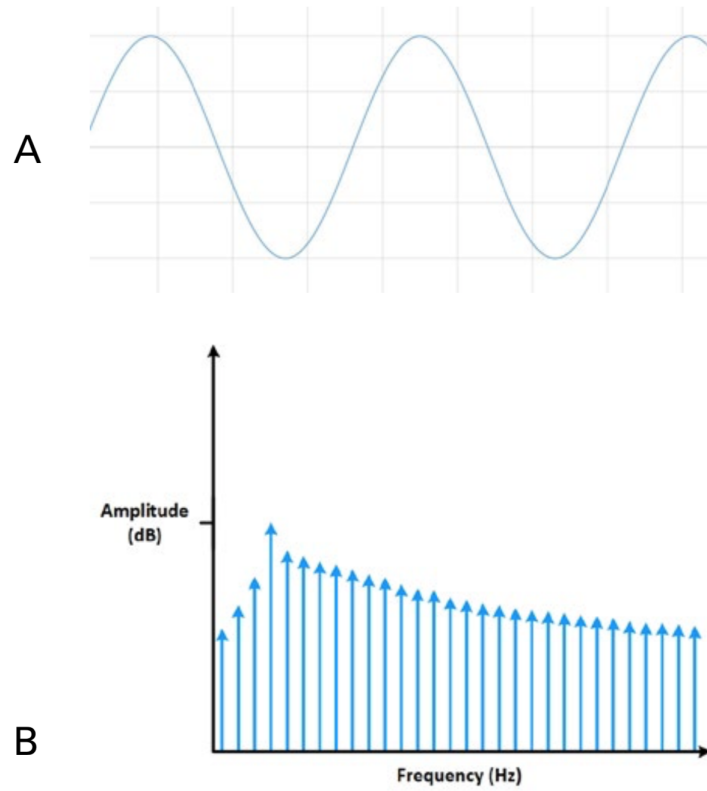
9.1.1 Spectrum Smearing

The Discrete Fourier Transform is an extremely useful tool, although it has limitations. When taking the Discrete Fourier Transform of a signal, the signal must be of a finite length, and the finite length signal is assumed to be periodic by the machinery of the DFT.

When taking the DFT of a signal, we are decomposing it into a sum of harmonics of the same period. For example, when taking the DFT of the following signal (assume discrete):



There is a single value in the frequency space because signal A (assumed to be periodic), is clearly made up of a sine wave of a certain frequency. Now, what happens if we take the DFT of the following signal (which is also assumed to be periodic):



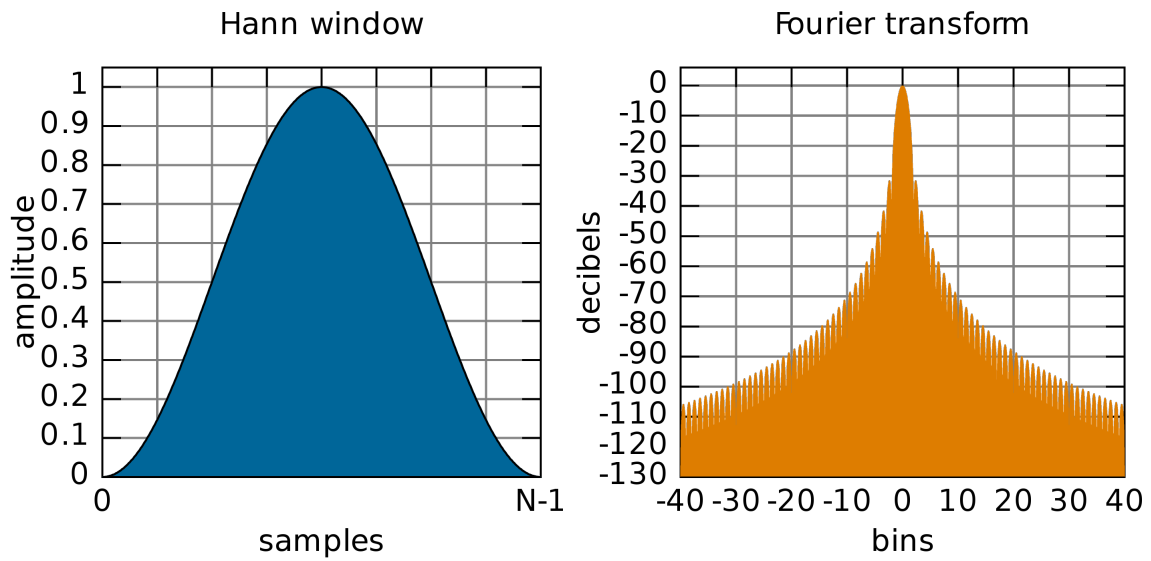
If the signal (A) is assumed to be periodic, and there is a non-integer number of periods of the signal wave present, the signal the DFT sees will be discontinuous! This new discontinuous signal is decomposed into harmonics that are continuous over the boundary, resulting in what is known as **spectrum smearing** or **spectral leakage**. The correct frequency is near the peak (but not properly captured by the DFT). The DFT is obviously not representative of the frequencies actually present in our signal.

9.1.2 Windowing

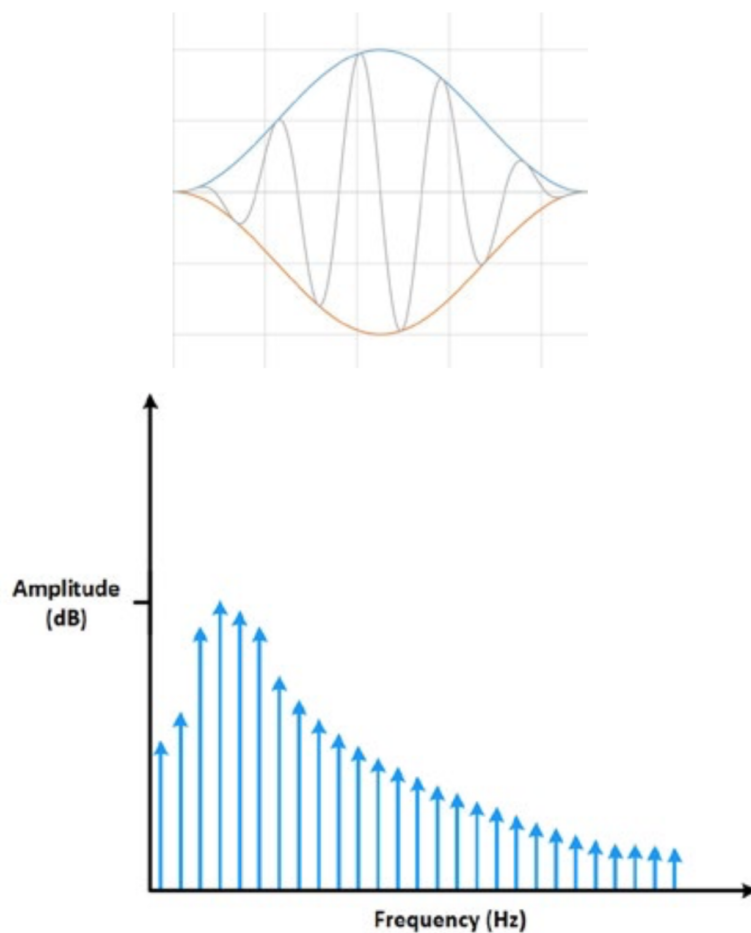
One possible method of dealing with the issue of spectral leaking intrinsic to the DFT is known as **windowing**. Windowing is the multiplying of the finite length signal of interest by a *window function*, which minimizes the effect of discontinuities.

One of the more common window functions is known as the Hanning Window, and takes the following form:

$$w[n] = \sin^2\left(\frac{\pi n}{N-1}\right) \quad (9.1)$$



Since this signal is zero valued at 0 and $N - 1$, and also has zero valued first derivative at 0 and $N - 1$, the affects of discontinuities on the frequency spectrum of the DFT are lessened. The effect of multiplying a signal by a Hanning is easily visualized by the following:



The unwanted frequencies are visibly reduced in power after the discontinuities are minimized.

9.2 Fast Fourier Transform

The method of calculating the DFT of a finite length signal is on the order of $O(n^2)$ when given simply as:

$$\hat{X}[k'] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi k' n/N} \quad (9.2)$$

This is very inefficient, and scales quickly as n increases. We can improve the efficiency by taking advantage of symmetries within the DFT. **Remember that the DFT is periodic in k , $X[k] = X[k + N]$.**

Suppose we assume our signal length is a power of 2, $N = 2^m$, and split up the DFT calculation into even and odd valued n as follows:

$$\hat{X}[k] = \underbrace{\sum_{n=0}^{N/2-1} x[2n] e^{-j2\pi k 2n/N}}_{\text{even } n} + \underbrace{\sum_{n=0}^{N/2-1} x[2n+1] e^{-j2\pi k (2n+1)/N}}_{\text{odd } n} \quad (9.3)$$

$$\hat{X}[k] = \underbrace{\sum_{n=0}^{N/2-1} x[2n] e^{-j2\pi k n/(N/2)}}_{\text{even } n} + e^{-j2\pi k/N} \underbrace{\sum_{n=0}^{N/2-1} x[2n+1] e^{-j2\pi k/(N/2)}}_{\text{odd } n} \quad (9.4)$$

We are in fact taking the length $N/2$ DFT of the even and odd parts of our original length N signal, and multiplying the odd part by a phase factor. Rewriting in simpler terms after denoting the even and odd valued DFTs as E and O respectively:

$$\hat{X}[k] = E[k] + e^{-j2\pi k/N} O[k] \quad (9.5)$$

Remember, since $E[k] = E[k + N/2]$ and $O[k] = O[k + N/2]$, only the phase factor has different values for $k < N/2$ and $k > N/2$. When applied recursively to calculate $E[k]$ and $O[k]$, we have significantly reduced the asymptotic runtime of the DFT to $O(n \log(n))$. You will write your own FFT algorithm and compare the speed to your original DFT as homework.

9.3 Time-Frequency Uncertainty Principle

One extremely important aspect of frequency analysis is known as the **time-frequency uncertainty principle**, (although many different variables have a finite uncertainty).

The most familiar uncertainty principle from quantum mechanics between a particle's position x and momentum p , formally stated as:

$$\sigma_x \sigma_p \geq \frac{\hbar}{2} \quad (9.6)$$

Where σ_x and σ_p are the standard deviations of the position and momentum of a particle respectively, and $\hbar$ is the reduced Planck constant. We will obviously not go into the derivation of this lower bound (take 8.04/8.05 to do that!).

We can take this familiar uncertainty principle and manipulate it until we have an uncertainty between time and frequency. In quantum mechanics, momentum is equal to $\hbar$ multiplied by angular spatial frequency:

$$p = \hbar \omega_s \quad (9.7)$$

Since the standard deviation scales in magnitude with the distribution of interest (taking X to be some distribution):

$$\sigma[X] = \sigma_x \iff \sigma[aX] = |a| \sigma_x \quad (9.8)$$

The quantum uncertainty relation between position and momentum can be rewritten as follows:

$$\sigma_x \hbar \sigma_{\omega_s} \geq \frac{\hbar}{2} \quad (9.9)$$

$$\sigma_x \sigma_{\omega_s} \geq \frac{1}{2} \quad (9.10)$$

We can *informally* convert the distributions of space x and spatial angular frequency ω_x to distributions of time, leaving us with the time-frequency uncertainty principle:

$$\boxed{\sigma_t \sigma_\omega \geq \frac{1}{2}} \quad (9.11)$$

Stated as follows: **the standard deviation of a signal as a function of time multiplied by the standard deviation of the same signal as a function of angular frequency is greater than or equal to $1/2$.**

This may seem very abstract, but this interactive visualization of a Gaussian signal may help your understanding.