# SFWR ENG 2MX3 Summary

Author: Kemal Ahmed
Instructor: Dr. MvM
Date: Winter 2014

*Math objects made using MathType; graphs made using Winplot.*

Please join GitHub and contribute to this document. There is a guide on how to do this on my GitHub.

CTRL-F (?) to find locations which need to fixed

## Table of Contents

## Systems

Order of writing symbols, such as exponents: digits-i-variables

There are 3 ways of representing a system:

- Difference Equation
- State Space Equations (ABCD), and
- Impulse Response

## Deltas

2 types of deltas (δ):

**Kronecker Delta**: discrete time domain, $\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & \text{else} \end{cases}, \int \delta(0) = 1$

**Dirac Delta**: continuous time domain, $\delta(t)$, $\delta(0) = \infty$, $\delta(\text{else}) = 0$, $\int \delta(0) = 1$

Also, Heaviside function is not defined at 0, whereas the step function is.

Discrete manipulates functions of n.
Continuous manipulates functions of t.

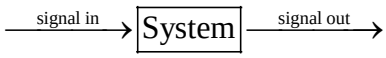
y = output
**Causal system**: system = 0 before impulse

Only use cosine curves; represent sine curves with a phase shift.

If you have a y(n−2) state, but not a y(n−1) state, include a y(n−1) state anyways.

**SISO**: single-input, single-output
**MIMO system**: Multiple Inputs Multiple Outputs.

All the systems we examine are "zero-state". Initial state is zero, s(0) = 0




## Discrete Systems

If there's no x in the equation, there must be a mistake because there's no input.

- More precise, more theoretical
- Discrete input signals
- Easier to process because it's digital

## Continuous System

- e.g. x(t) = 2cos(t)
- Analog

# Difference equation

## Definition

The first sum is manipulation of input signals and the second part is manipulation of output signals.

$y(n)=\sum_{k=0}^{N}\alpha_k x(n-k)+\sum_{k-1}^{M}\beta_k y(n-k)$, which is interchangeable with:

$$H(\omega)=\frac{\sum_{k=0}^{\infty}\alpha_k e^{-i\omega k}}{1-\sum_{k=1}^{\infty}\beta_k e^{-i\omega k}}$$ (frequency response)

# State Space Equations

## ABCD

The **state space equations method** is also known as **ABCD Method**. This is because you represent your states by the four matrices, **A**, **B**, **C**, and **D**. There are 2 state space equations:

- Next state equation
- Output

Given *k* states, **next state equation**:

$$\mathbf{S}(n+1)=\begin{bmatrix}S_1(n+1)\\S_2(n+1)\\\vdots\\S_k(n+1)\end{bmatrix}=\mathbf{A}\begin{bmatrix}S_1(n)\\S_2(n)\\\vdots\\S_k(n)\end{bmatrix}+\mathbf{B}x(n)$$

Notice how the left side of the equation is the next state of each state? Expanding the matrices would show you how many of each state and the input each state represents.

The output equation is just re-writing the given equation in terms of the states you determined and your inputs:

$$y(n)=\mathbf{C}\begin{bmatrix}S_1(n+1)\\S_2(n+1)\\\vdots\\S_k(n+1)\end{bmatrix}+\mathbf{D}x(n)$$

The state space equation method use matrices of states to represent any equation in a concise way.

States are any part of the given equation that is not y(n) or x(n). That is because the number of x(n)'s (i.e. inputs) are represented by the **B** matrix. On the other hand, the y(n) is not considered a state, but rather, a collection of states. Thus, if your next state is y(n), expand it by equating it to the initial equation of y(n).

**e.g.)**

$$y(n) = x(n) - y(n-2)$$

Recall that x(n) is already going to be represented by the **B** matrix, so it cannot be a state. However, the $y(n-2)$ *can* be represented by a state. Let's call it $S_1$:

$$S_1(n) = y(n-2)$$

Now, the next state will be:

$$S_1(n+1) = y(n+1-2) = y(n-1)$$

Hey! There's no matrix for y(n − 1). There are only matrices for y(n) and x(n)! So we should make another state. Think of it like a temporary variable where the value of the variable is passed down, until $S_1$.

Ok, so let's call this state $S_2$:

$$S_2(n) = y(n-1)$$

Let's look at the next states of both $S_1$ and $S_2$:

$$S_1(n) = y(n-2) \qquad S_1(n+1) = y(n-1)$$

$$S_2(n) = y(n-1) \qquad S_2(n+1) = y(n)$$

Remember that we're supposed to represent the next states by a combination of the previous states? So let's change those next states. Recall: when you have y(n), you expand it into the states it represents.

$$S_1(n+1) = y(n-1) = S_2(n)$$

$$S_2(n+1) = y(n) = x(n) - S_1(n)$$

For the next part, you need to understand matrix expansion. For now, I'll re-write each next-state equation individually before combining it to show what's actually going on:

$$\begin{aligned} S_1(n+1) &= 0S_1(n) + 1S_2(n) + 0x(n) \\ &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} S_1(n) \\ S_2(n) \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} x(n) \end{aligned}$$

$$\begin{aligned} S_2(n+1) &= -1S_1(n) + 0S_2(n) + 1x(n) \\ &= \begin{bmatrix} -1 & 0 \end{bmatrix} \begin{bmatrix} S_1(n) \\ S_2(n) \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} x(n) \end{aligned}$$

Now put those two together to get:

$$\mathbf{S}(n+1) = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} S_1(n) \\ S_2(n) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{B}} x(n)$$

The output equation is found the same way:

$$y(n) = -1S_1(n) + 0S_2(n) + x(n)$$
$$= \underbrace{\begin{bmatrix} -1 & 0 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} S_1(n) \\ S_2(n) \end{bmatrix} + \underbrace{[1]}_{\mathbf{D}} x(n)$$

You can use these matrices for determining impulse response.

**Linear Time Invariant** (LTI) **System**: you may flip order of system and delay (?)

## Converting to Difference Equation

Euler's equations convert states to the difference equation.

$$e^{i\theta} = \cos\theta + i\sin\theta$$
$$\cos\theta = \tfrac{1}{2}\left(e^{i\theta} + e^{-i\theta}\right)$$
$$\sin\theta = \tfrac{1}{2i}\left(e^{i\theta} - e^{-i\theta}\right)$$

## Compound Interest

An application of the state space equation other than signals is for determining compound interest.

$n$: billing period
$\alpha$: interest
$x$: deposits

$$y(n) = y(n-1) + \alpha y(n-1) + x(n)$$
$$= (1+\alpha)y(n-1) + x(n)$$

Similar to an impulse, let's model someone putting money in a bank and leaving it there forever, where $A$ is the value of an initial deposit:

$$x(n) = \begin{cases} A, & n = 0 \\ 0, & \text{else} \end{cases}$$
$$y(0) = A$$
$$y(1) = (1+\alpha)A$$
$$y(2) = (1+\alpha)^2 A$$

So you can interpolate: $y(n) = (1+\alpha)^n A$

# Impulse Response

Impulses can be modeled by the Kronecker delta function.

## FIR

**Finite Impulse Response** (FIR) system: A system that has an impulse response that has a finite duration (is zero at a finite time). One way of seeing if it is finite is if it has no y's in it,

### e.g.) State Space Table

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

How we model systems in this course is using a **state space table**. This table analyzes what the output of the given system is when you are given an impulse. The prof usually leaves the labels blank, but this time, I'll leave them in there. Next time, however, they won't be there:

| Time | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Input [$x(n)$] | 1 | 0 | 0 | 0 | 0 |
| Output [$y(n)$] | 1 | 0 | 1 | 0 | 0 |

From this table, you can tell that it's an FIR system, since the output is 0 after $t = 2$.

### IIR

**Infinite Impulse Response** (IIR) system. A system that has an impulse response that has an infinite duration (continues to respond indefinitely). IIR systems generally have $y$ on both sides of the equation

### e.g.)

$y(n) = x(n) + 0.5y(n-1)$

| | 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|
| | 1 | ½ | ¼ | ⅛ | 1/16 |

Notice how the system doesn't end? This is why it is called an IIR system. Also, notice how the labels are gone? Get used to it and imagine them there every time you make a table like this.

## Impulse Response

**Impulse Response** is notated as $h(n)$. It is the output, $y(n)$, when the input, $x(n)$ is a combination of shifts of the Kronecker delta function.

To clarify what the equations mean: $\xrightarrow{x(n)} \boxed{h(n)} \xrightarrow{y(n)}$

Remember ABCD representation? Impulse response can be represented by inputting the ABCD matrices:

$$h(n) = \begin{cases} 0, & n < 0 \\ \mathbf{D}, & n = 0 \\ \mathbf{CA}^{n-1}\mathbf{B}, & n > 0 \end{cases}$$

## Convolution Sum

$$y = h * x$$

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

If something appears to have the convolution equation, it will be convolution and thus, you may use its properties.

## Impulse response to difference equation

Sometimes your impulse response may be specified. There are 2 ways of converting your impulse response into the difference equation:

- Convolution Sum method
- Table method

### e.g.)

We're going to use the same example to demonstrate both methods

$$h(n) = \begin{cases} 0, & \text{else} \\ 1, & n = 0 \\ 2, & n = 1 \\ 3, & n = 2 \end{cases}$$

$x(n) = \delta(n) + \delta(n-2)$

To simplify the process, take the $h(n)$ and split it up into a sum of diracs:

$$h(n) = \delta(n) + 2\delta(n-1) + 3\delta(n-2)$$

### Convolution Equation method

$$\boxed{y(n) = \sum_{i=-\infty}^{\infty} h(n-i)\,x(i)}$$

Let's plug in some values of $n$:

$$y(0) = \sum_{i=0}^{\infty} h(0-i)\,x(i)$$

However, since $h$(anything < 0) = 0, we only get output when $i = 0$:

$$\begin{aligned} y(0) &= h(0)\cdot x(0) \\ &= 1\cdot 1 = 1 \end{aligned}$$

$$\begin{aligned} y(1) &= \sum_{i=0}^{\infty} h(\underbrace{1-i}_{0,1})\,x(i) \\ &= h(1-0)\cdot x(0) + h(1-1)\,x(1) \\ &= (2\cdot 1) + (1\cdot 0) = 2 \end{aligned}$$

$$\begin{aligned} y(2) &= \sum_{i=0}^{\infty} h(2-i)\,x(i) \\ &= h(2-0)\,x(0) + h(2-1)\,x(1) + h(2-2)\,x(2) \\ &= (3\cdot 1) + (2\cdot 0) + (1\cdot 1) \\ &= 3 + 0 + 1 = 4 \end{aligned}$$

$$y(3)=\sum_{i=0}^{\infty}h(3-i)x(i)$$
$$=h(3-0)x(0)+h(3-1)x(1)+h(3-2)x(2)+h(3-3)x(3)$$
$$=(0\cdot 1)+(3\cdot 0)+(2\cdot 1)+(1\cdot 0)$$
$$=2$$

We're supposed to stop at this point for some reason.

## Table method

For each of the impulses in $x(n)$, look at the impulse response. Add all the $h(n)$'s up at the end.

$$\boxed{y(n)=\sum_{k=-\infty}^{\infty}h(k)x(n-k)}$$

| n | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| δ(n) [x(0)] | 1 | 0 | | 0 | 0 | 0 |
| h(0) | 1 | 2 | 3 | 0 | 0 | 0 |
| δ(n−2) | 0 | 0 | 1 | 0 | 0 | 0 |
| h(2) | 0 | 0 | 1 | 2 | 3 | 0 |
| y(n) | 1 | 2 | 4 | 2 | 3 | 0 |

## Kemal's Method

Someone else probably thought of this before me, but there's a faster way of directly converting your impulse response to your difference equation:

$$h(n)=\begin{cases}\alpha_0, & n=0\\ \vdots & \\ \alpha_N, & n=N\end{cases}$$

Plug it into: $y(n)=\sum_{k=0}^{N}\alpha_k x(n-k)$ and expand.

It may be easier to convert the impulse response to an equation of dirac deltas.

If you aren't given an equation for $x$ at this point, assume $x(n)=\delta(n)$. You may then leave it in/convert to dirac delta equations.

## Step Response

**Step Response**: $\text{step}(n-k)=\begin{cases}0, & n<k\\ 1, & n\geq k\end{cases}$

$$y=h*x$$

$$y(n)=\sum_{k=-\infty}^{\infty}h(k)\,\text{step}(n-k)$$

You can't have $h(k<0)$, since time starts at 0:

$$y(n)=\sum_{k=0}^{\infty}h(k)\,\text{step}(n-k)$$

k can only be as big as n before that iteration of the sum is 0:

$$y(n)=\sum_{k=0}^{n}h(k)$$

## Discrete and Continuous Frequency

| | Discrete | Continuous |
|---|---|---|
| **Period** | $p=\frac{\text{samples}}{\text{cycle}}$ (must be a natural number) | $p=\frac{\text{seconds}}{\text{cycle}}$ |
| **Frequency** | $f=\frac{1}{p}=\frac{\text{cycles}}{\text{sample}}$ | $f=\frac{1}{p}=\frac{\text{cycles}}{\text{second}}=\text{Hz}$ |
| **Normalized Frequency** | $\omega=2\pi f=\frac{\text{radians}}{\text{sample}}$ | $\omega=2\pi f=\frac{\text{radians}}{\text{second}}$ |
| **Sampling Frequency** | $f_s=\frac{\text{samples}}{\text{second}}$ | |
| **Pitch** | $\text{Pitch}=f_s\cdot f_{\text{discrete}}=\frac{\text{samples}}{\text{second}}\cdot\frac{\text{cycles}}{\text{sample}}=\frac{\text{cycles}}{\text{second}}=\text{Hz}$ | |

Remember that *upper-case* variables (frequency-domain) are *multiplied*: $Y(\omega)=H(\omega)\times X(\omega)$, and *lower-case* variables (time-domain) are *convoluted*: $y(t)=h(t)*x(t)=\int_{-\infty}^{\infty}x(t-\tau)h(\tau)\,\mathrm{d}\tau$

$$\begin{aligned}x(n)&=k(n+kp)\\&=\cos\left(\tfrac{\pi}{4}n\right)\leftarrow\omega=\tfrac{\pi}{4}\\&=\tfrac{\pi}{4}n=k2\pi\\&\Rightarrow p=8\end{aligned}$$

Your period is selected such that the coefficient × p = a multiple of 2π. If there is a phase shift, don't worry about it when calculating period, frequency, or normalized frequency.

Often the coefficient can actually be your normalized frequency (remember this to save steps). Choose an $\omega_0$, such as $\omega_0=\pi/4$ and find the period, such that $\omega_0\cdot p$ is any multiple of 2π, i.e. $\omega_0\cdot p = k2\pi$

**e.g.**

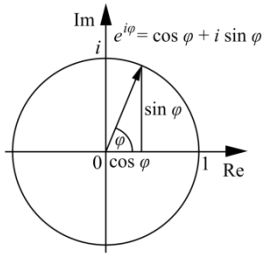
If you wanted to normalize $x=\frac{3\pi}{4}n$ (should be divisible by 2)

$p = 8$

$\omega = \frac{2\pi}{p} = \frac{\pi}{4}$

## Complex Numbers

Draw a graph where the x values represent the real dimension and the y values represent the complex dimension. This is actually the graph of $e^{ix}$ (pretend the φ's are actually x's):




Refer to Euler's equations a lot (in the difference equations section).

e.g. $e^{i\frac{\pi}{4}}$

**argument** ($\measuredangle$): the degree of the point on the circle
The argument of each point is actually the value of x currently being examined, such that
$\measuredangle e^{ix} = x$
$\measuredangle e^{i\frac{\pi}{4}} = \frac{\pi}{4}$
**gain**: magnitude of the line connecting the origin to the point

Divide by the magnitude on the unit circle. Thus, the point is: $\left(\frac{1}{2}, \frac{i}{2}\right)$
Know your special triangles.
$\sqrt{1^2 + 1^2} = \sqrt{2}$

Also remember how the unit circle works, such as how $e^{i\pi} = -1$

Summary: numbers can be converted to their complex exponential equivalent by finding the gain and the angle, then num = gain $e^{i\measuredangle}$

$\frac{1}{i} = -i$

## Frequency Response

### Intro

**Frequency Response** [H(ω)]: Although H(ω) is actually called the transfer function (if you ever need to Google it), call it frequency response for now

Upper-case variable: frequency domain (i.e. a function of ω)
Lower-case variable: time domain (i.e. a function of t or n)

Formula: $H(\omega)=\frac{Y(\omega)}{X(\omega)}=\frac{\mathcal{L}\{y(t)\}}{\mathcal{L}\{x(t)\}}=\frac{\sum_{k=0}^{\infty}Y_k e^{-i\omega_0 k}}{1-\sum_{k=1}^{\infty}X_k e^{-i\omega_0 k}}$

e.g. if $x(n)=e^{i\omega n}$, $y(n)=H(\omega)e^{i\omega n}$

$\measuredangle H(\omega)$ = phase shift of the cosines in H(ω)

The gain is represented by $\|H(\omega)\|$

When your $X_k$'s go to 0, they are known as **poles** or **eigenfrequencies** because they result in an ∞ frequency response.
Eigenfrequency is the frequency that will cause a glass to break
**Resonance frequency**

**Conservative**: you don't retain energy (?)

## Discrete LTI Signal

Given a discrete-time signal, x(n) and an H(ω), the output is computed in the following way:
Your input may contain multiple terms. Treat each cosine individually.

To understand what variables I am referring to, take each cosine and compare it to this general equation: $\boxed{y(n)=\|H(\omega)\|\cos(\omega n+\measuredangle H(\omega))}$

Note: if x(n) = a number [such as 5], treat it as if k = 0, so x(n) = 5cos(0n)

But wait! Was that a y on the left side of the equation? Shouldn't that be an x? No. I just showed you the output equation. This is the equation you're going to put your answer in after you have solved for the variables. I showed you this first because your x is going to be very similar to this.

1. Convert all sines in your x(n) to cosines
2. Identify the ω's from each of the equations

To make it easier to organize the information you have just gotten, make a table with a different row for each cosine. This works because each cosine should have its own ω. To find the H(ω) for each row, plug the ω of that row into H(ω)

**e.g.**

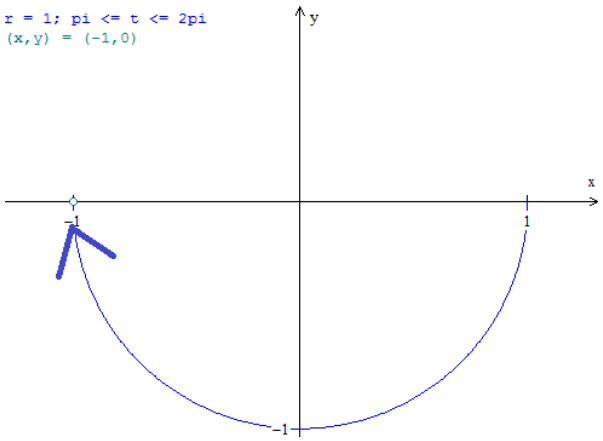
$$H(\omega)=\cos(\omega)$$

$$x(n)=2+\cos\left(\tfrac{\pi}{2}n+\tfrac{\pi}{4}\right)+\sin\left(\pi n+\tfrac{3\pi}{2}\right)$$

$$=2+\cos\left(\tfrac{\pi}{2}n+\tfrac{\pi}{4}\right)+\cos(\pi n+\pi)$$

| ω | H(ω) | ‖H(ω)‖ | $\measuredangle H(\omega)$ |
|---|---|---|---|
| 0 | cos(0) = 1 | 1 | 0 ←arg($e^{i0}$) |
| π/2 | cos(π/2) = 0 | 0 | – |
| π | cos(π) = −1 | 1 | −π |

The argument will always be negative because you can't look ahead of time.




Now that we have this information, plug it into the y-equation from earlier.

## e.g. Find y using FR

Find the output to this system

y(n) + y(n−1) = 2x(n) − 5x(n−2)

Convert to frequency response

$$H(\omega)e^{i\omega n}+H(\omega)e^{i\omega(n-1)}=2e^{i\omega n}-5e^{i\omega(n-2)}$$

$$H(\omega)\left(e^{i\omega n}+e^{i\omega n}e^{-i\omega}\right)=2e^{i\omega n}-5e^{i\omega n}e^{-2i\omega}$$

$$H(\omega)\left(1+e^{-i\omega}\right)=2-5e^{-2i\omega}$$

Factor everything out to isolate H(ω):

$$H(\omega)=\frac{2-5e^{-2i\omega}}{1+e^{-i\omega}}$$

Use the frequency response to determine the output

$$\boxed{H(\omega)=\|H(\omega)\|e^{i\measuredangle H(\omega)}}$$

| ω | H(ω) | ‖H(ω)‖ | $\measuredangle H(\omega)$ |
|---|---|---|---|
| 0 | − 3/2 | 3/2 | −π |

| | | | |
|---|---|---|---|
| π/2 | Just plugin ω into H<br>$\frac{2-5e^{-2i\frac{\pi}{2}}}{1+e^{-i\frac{\pi}{2}}}$<br>$=\frac{2-5(-1)}{1-i}$<br>$=\frac{7}{1-i}$<br>$=\frac{7+7i}{1+1}$<br>$=\frac{7}{2}+\frac{7}{2}i$ | What is the magnitude of the point, H(ω), on the $e^{ix}$ graph<br>$\sqrt{3.5^2+3.5^2}=3.5\sqrt{2}$ | $\frac{7}{2}+\frac{7}{2}i=3.5\sqrt{2}e^{i\sphericalangle H(\omega)}$<br>$1+i=\sqrt{2}e^{i\sphericalangle H(\omega)}$<br>$\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i=e^{i\sphericalangle H(\omega)}$<br>$\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i=e^{i\sphericalangle H(\omega)}$<br>$\sin^{-1}\left(\frac{1}{\sqrt{2}}\right)=\frac{\pi}{4}\boxed{=\frac{-7\pi}{4}}$ |
| π | 0 | 0 | 0 |

Where ϕ is the previous phase shift:

$$\boxed{y(n)=\|H(\omega)\|[\cos/\sin](\omega n+\phi+\sphericalangle H(\omega))}$$

$$y(n)=\tfrac{3}{2}\cos(-\pi)=\tfrac{-3}{2}+5\cos\left(\tfrac{\pi}{2}n-\tfrac{7\pi}{4}+\tfrac{\pi}{4}\right)$$

$$y(n)=\tfrac{-3}{2}+\tfrac{7}{\sqrt{2}}\cos\left(\tfrac{\pi}{2}n-\tfrac{3\pi}{2}\right)$$

# Fourier

## Impulse Response to Frequency Response

In the Fourier Series, $\omega$ is not what it usually is ($2\pi f$), since it changes with each $k$ or $n$ (depending on discrete / continuous). That is why we represent $2\pi f$ by a constant, $\omega_0$. It is calculated the same way as you would calculate ω, normally (i.e. $\omega_0 = 2\pi/p = 2\pi$f).

[$\omega_0$]:
For discrete frequency: $\omega = \omega_0 \times k$
For continuous frequency: $\omega = \omega_0 \times n$ [or $m$, depending on your notation]

**Forward transform**: time → frequency domain (look for negative exponents on e)
Transform$^{-1}$: frequency → time domain (look for positive exponents on e)

If $h$ is a function of time, then CTFT; if not, DTFT. i.e. h(t), do CTFT; h(n), do DTFT
The impulse response, $h(n)$, of a system is the $y(n)$ when $x(n)$ is an impulse: $\delta(n)$.

$k$ is the specific frequency you are aiming to get. It ranges from 0 to (p−1)

N = p

## Master Chart

Note: DFT/FFT is below

| | FS (Fourier Series) | FT (Fourier Transform) |
|---|---|---|
| DT (Discrete Time) | $X_k = \frac{1}{p}\sum_{m=0}^{p-1} x(m)e^{-imk\omega_0}$<br>$x(n) = \sum_{k=0}^{p-1} X_k e^{ik\omega_0 n}$ | $X(\omega) = \sum_{m=-\infty}^{\infty} x(m)e^{-i\omega m}$<br>$x(n) = \frac{1}{2\pi}\int_0^{2\pi} X(\omega)e^{i\omega n}\mathrm{d}\omega$<br>See more below |
| CT (Continuous Time) | $X_n = \frac{1}{p}\int_0^p x(t)e^{-in\omega_0 t}\mathrm{d}t$<br>$x(t) = \sum_{n=-\infty}^{\infty} X_n e^{in\omega_0 t}$<br>See more below | $X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}\mathrm{d}t$<br>$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{i\omega t}\mathrm{d}\omega$<br>Summary:<br>$\mathrm{CTFT}\left(ce^{i\omega_0 t}\right) = c\cdot 2\pi\delta(\omega-\omega_0)$<br>**essentially, Laplace** |

## CTFS

Notice how there is no range for the CTFS? That's because it is actually from $-\infty$ to $\infty$. An easier way to do it than computing it is to convert everything from your x(t) equation into complex exponentials with the form $x(t) = e^{i\omega t}$. Now realize that $X_n$ is the output of impulse response, so

$$y(t) = H(\omega)x(t)$$

$$H(\omega) = \frac{y(t)}{x(t)} = \frac{X_n}{e^{i\omega t}} = X_n e^{-i\omega t} = X_n e^{-i\omega_0 nt}$$

That means the coefficients in front of each of your complex exponentials in your x(t) equation represent a different $X_n$.

e.g. $x(t) = \frac{1}{2}e^{it} + \frac{1}{2}e^{-it} = \frac{1}{2}e^{i\cdot 1\cdot 1\cdot t} + \frac{1}{2}e^{i\cdot 1(-1)t}$, $X_{n=1}=X_1 = ½$, $X_{n=-1}=X_{-1} = ½$.

Notice how *n* is directly related to ω? That's why each *n* is known as a frequency.

Sometimes, you can notice that the ω's are multiples of the same ω. This is because the function is periodic.

### e.g.)

$\omega_1 = \frac{\pi}{4}, \omega_2 = \frac{\pi}{2}$

CTFS: $X_n = \frac{1}{p}\int_0^p x(t)e^{-i\omega_0 nt}\mathrm{d}t$

Taking $X_n = \frac{1}{p}\int_0^p \left(\frac{1}{2}e^{i\frac{\pi}{2}t} + \frac{1}{2}e^{-i\frac{\pi}{2}t} + \frac{1}{2}e^{i\frac{\pi}{4}t} + \frac{1}{2}e^{-i\frac{\pi}{4}t}\right)e^{-i\omega_0 nt}\mathrm{d}t$,

Choose an $\omega_0$, such as $\omega_0 = \pi/4$ and find the period, such that $\omega_0 \cdot p$ is any multiple of $2\pi$, i.e. $\omega_0 \cdot p = k2\pi$

$p = 8$

$$X_n = \frac{1}{16}\int_0^8 \left(e^{i\frac{\pi}{2}t} + e^{-i\frac{\pi}{2}t} + e^{i\frac{\pi}{4}t} + e^{-i\frac{\pi}{4}t}\right)e^{-i\frac{\pi}{4}nt}\mathrm{d}t$$

Now, put all the items in terms of $\omega_0$:

$$X_n = \frac{1}{16}\int_0^8 \left(e^{i\frac{\pi}{4}2t} + e^{-i\frac{\pi}{4}2t} + e^{i\frac{\pi}{4}t} + e^{-i\frac{\pi}{4}t}\right)e^{-i\frac{\pi}{4}nt}\mathrm{d}t$$

Now, expand:

$$X_n = \frac{1}{16}\int_0^8 e^{i\frac{\pi}{4}t(1-n)} + e^{i\frac{\pi}{4}t(2-n)} + e^{i\frac{\pi}{4}t(1+n)} + e^{i\frac{\pi}{4}t(2+n)}\mathrm{d}t$$

$$\boxed{\begin{gathered}\frac{1}{p}\int_0^p x(t)e^{-i\omega_0 kt}\mathrm{d}t = \frac{p}{p}x(t)\delta(k) \\ \Rightarrow \frac{1}{p}\int_0^p e^{-i\omega_0 kt}\mathrm{d}t = \delta(k)\end{gathered}}$$

$$\begin{aligned}X_n &= \frac{1}{16}\int_0^8 e^{i\frac{\pi}{4}t(1-n)}\mathrm{d}t + \frac{1}{16}\int_0^8 e^{-i\frac{\pi}{4}t(1+n)}\mathrm{d}t + \frac{1}{16}\int_0^8 e^{i\frac{\pi}{4}t(2-n)}\mathrm{d}t + \frac{1}{16}\int_0^8 e^{-i\frac{\pi}{4}t(2+n)}\mathrm{d}t \\ &= \tfrac{1}{2}\delta(1-n) + \tfrac{1}{2}\delta(1+n) + \tfrac{1}{2}\delta(2-n) + \tfrac{1}{2}\delta(2+n)\end{aligned}$$

$$X_1 = \tfrac{1}{2}, X_{-1} = \tfrac{1}{2}, X_2 = \tfrac{1}{2}, X_{-2} = \tfrac{1}{2}$$

## CTFT/DTFT

$$\delta(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}\mathrm{d}\omega$$

$$\int_{-\infty}^{\infty} e^{i\omega t}\mathrm{d}\omega = 2\pi\delta(\omega)$$

$$\boxed{\int_{-\infty}^{\infty}\left(c\cdot e^{i\omega_0 t}\right)\mathrm{d}t = c\cdot 2\pi\delta(\omega-\omega_0)}$$

$$\boxed{\int_{-\infty}^{\infty}\left(c\cdot e^{i\omega_0 n}\right)\mathrm{d}n = c\cdot 2\pi\delta(\omega-\omega_0)}$$

Since sine and cosine are continuous functions, they can't simply be represented by impulses. Thus, assume 2π-periodicity and add '−2πn' to the end of the Kronecker Delta:

$$\cos(\omega_0 t) \Rightarrow \pi\sum_{n=0}^{7}\left[\delta(\omega+\omega_0-2\pi n) + \delta(\omega-\omega_0-2\pi n)\right]$$

$$\sin(\omega_0 t) \Rightarrow \pi i\sum_{n=0}^{7}\left[\delta(\omega+\omega_0-2\pi n) - \delta(\omega-\omega_0-2\pi n)\right]$$

## DFT/FFT

The DFT is actually the most popular transform, because there's an efficient algorithm for it called the **Fast Fourier Transform** (FFT). It's the same as the DTFS, but with different scaling. See the difference:

| DTFS | DFT/FFT |
|---|---|

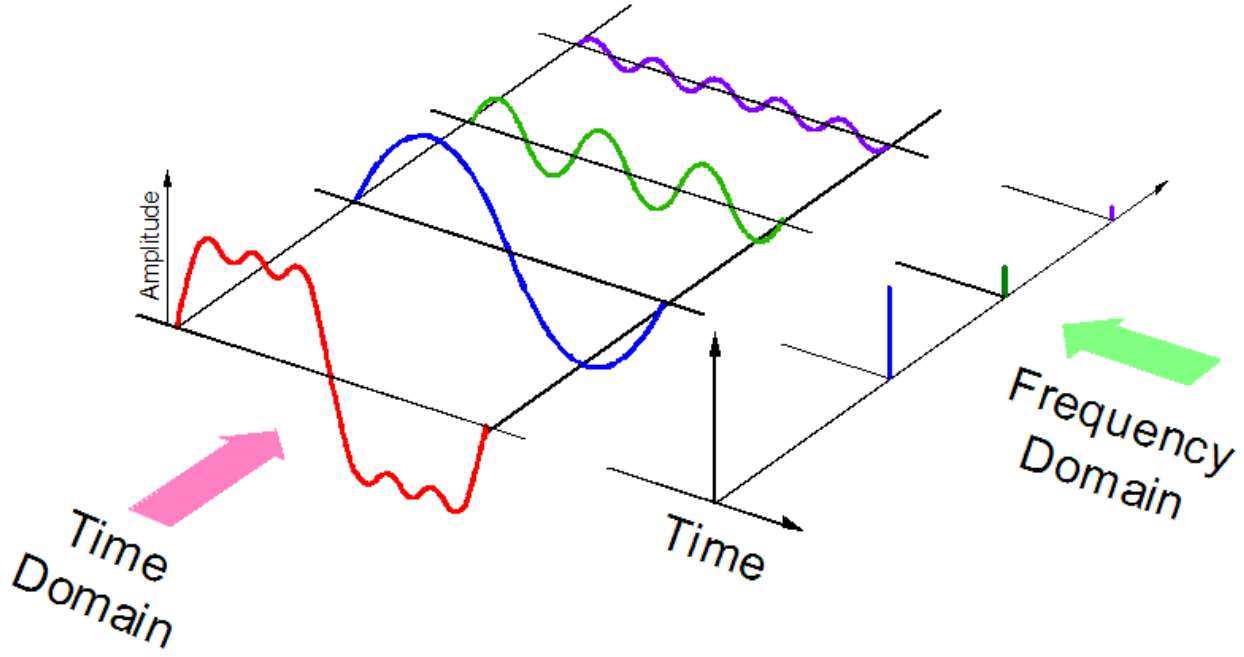
| | |
|---|---|
| $X_k = \frac{1}{p}\sum_{m=0}^{p-1} x(m) e^{-imk\omega_0}$ <br> $x(n) = \sum_{k=0}^{p-1} X_k e^{ik\omega_0 n}$ | $X_k = \sum_{m=0}^{p-1} x(m) e^{-imk\omega_0}$ <br> $x(n) = \frac{1}{p}\sum_{k=0}^{p-1} X_k e^{ik\omega_0 n}$ |

The DFT$^{-1}$ is also known as the **synthesis equation**.

Because the [forward] DFT doesn't divide out the number of points, *p*, using a higher sampling rate will make the DFT larger for the same input. This is why it's actually better to scale the DFT output by 1/p anyways when looking at it afterwards.

### Time and Frequency Domain

Gif demonstration of what the frequency domain is.




## Filter a signal

When designing a filter, use tilde ( $\tilde{H}$ ) to denote a prototype of your filter.

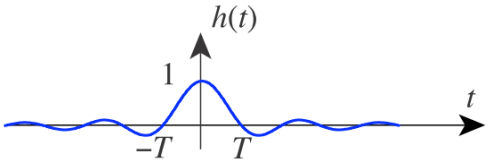
Note: “removal of frequency component at a point” means H(point) = 0, so you just pretend the value is a root at that time.

**Normalize**:

### Sinc Function

Filtered signals are generally represented by the **sinc function**: $y(x) = \frac{\sin(x)}{x}$

**Normalized sinc function**: $y(x)=\frac{\sin(\pi x)}{\pi x}$




Fourier of sinc:

$$\forall\, \omega \in \mathbb{R}, \quad X(\omega) = \begin{cases} T & \text{if } |\omega| \le \pi/T \\ 0 & \text{otherwise} \end{cases}$$

## e.g. Notch filter

a.k.a. band-stop filter

$H\left(\frac{\pi}{2}\right)=0$

$H(0)=1$

1) Set up the equation such that you have $\tilde{H}(\omega)=\prod\left(e^{-i\omega}-e^{-i\{\text{roots}\}}\right)$. In our case, your only root is at π/2, so:

$\tilde{H}(\omega)=\left(e^{-i\omega}-e^{-i\frac{\pi}{2}}\right)$

2) These systems require **Conjugate complex symmetry (CCS)** (When do things require CCS?) You need it so that your answer is real. If you don't have it, your output will be incorrect.

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

To allow for conjugate complex symmetry, we'll add a second part to the prototype:

$$\tilde{H}(\omega)=\left(e^{-i\omega}-e^{-i\frac{\pi}{2}}\right)\underbrace{\left(e^{-i\omega}-e^{i\frac{\pi}{2}}\right)}_{\text{allows CCS}},$$

$$=e^{-i2\omega}+1$$

To account for this, add a second initial condition, the complex conjugate, $H\left(\frac{-\pi}{2}\right)=0$.

3) Now plug in your other initial condition:

Although $H(0)=1, \tilde{H}(0)=2$.

To account for the difference, insert a constant of ½:

$H(\omega)=\frac{1}{2}\left(e^{-i2\omega}+1\right)$

## Types of Filters

http://www.falstad.com/fourier/

**Sawtooth signals** are periodic, so you have to use the Fourier Series on it.

$$x(t)=\frac{a_0}{2}+\sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi t}{1}\right)+b_n \sin\left(\frac{n\pi t}{1}\right)$$

## Filter Design

# Sampling

**sampling**: $x(t) \xrightarrow{\text{sampling}} x(n)$; think Riemann sum

Unfortunately, when you do sampling, you lose information because you're taking the value at each point and storing it in digital form.

**sampling period** (a.k.a. sampling interval) [Δt]: time between signals

**sampling frequency** (a.k.a. sampling rate): 1/Δt, Hz

[ω] (radians/second): 2π×f

Sample a continuous signal into a discrete signal by plugging the equation of the continuous signal into the discrete

$x(n) = x(\Delta t \cdot n)$,

$x(t) = \cos(\omega_1 t) + \cos(\omega_2 t)$

$x(n) = \cos(\omega_1 \cdot \Delta t \cdot n) + \cos(\omega_2 \cdot \Delta t \cdot n)$

Impulse contains all frequencies evenly.

If you are sampling a given frequency band, $p = \frac{f_s}{f_{\text{band}}}$

# Z and Laplace Transform

If you aren't given a value for x, assume it's a delta function.

e.g. Practice 3, question 3 $\ddot{y} + 4\dot{y} - 2y = x$

1) Laplace:

$s^2 Y(s) + 4sY(s) - 2Y(s) = X(s)$

2) Isolate for Y:

$Y(s) = \frac{1}{s^2 + 4s - 2} X(s)$

Poles of the transfer function are the coefficients of X(s) function.
Since you assume x(t) = δ(t), so X(s) = 1

3)

$$H(s) = \frac{1}{s^2 + 4s - 2} \cdot 1$$

4) Quadratic equation OR factor to find the poles

$$= \frac{1}{\left(s - \left(-2 + \sqrt{6}\right)\right)\left(s - \left(-2 - \sqrt{6}\right)\right)}$$

This region of convergence **doesn't** include the imaginary axis.

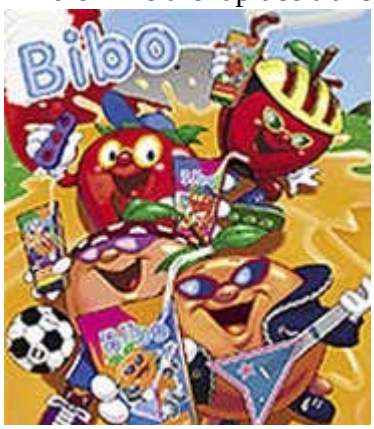
The region of convergence of *H*(*s*) must include the imaginary axis in order for the system to be stable.

*H*(*s*) is:

1. Called the **transfer function**, and is
2. The Laplace transform of the impulse response, h(t).

### Z-transform

Kind of like the laplace transform but for discrete functions.




## Block Diagrams

⊕ ALWAYS has 2 inputs, one output. If anything else, ignore. Output = sum of inputs.
When a wire splits up into 2, multiply outputs of the junction by the input.

When a wire encounters a delay or an amplifier or anything without a junction, multiply your current value by the object.

Feedback loop: jump ahead, then look back. Also,

$$H(\omega) = \frac{H_1(\omega)}{1 - H_1(\omega)H_2(\omega)}$$

D = delay = $e^{-i\omega}$
*Bonus*: D is technically a function, so keep it on the left of things that depend on it, similar to d/dt