ECE, Signal Processing

RustColeone

October 28, 2021

Contents

1 Sinusoidal Signals

1. Sinusoidal signals are the basic building blocks in the theory of signals & systems
2. Arise as soutions to differential equations that through the laws of physics, describe common physical phenomena

1.1 Mathematical Formula

let a signal be $s(t)$, where t represents the time t of a given time, we have:

$$s(t) = Acos(2\pi f_0 + \phi) \tag{1}$$

A linear function of time t, in which **A is the amplitude** that scales the cosine signal in y dimention, since

$$-1 \leq cos(x) \leq 1 \text{for all } x, s(t) \in [-A, +A] \tag{2}$$

ϕ **is the phase in (rad)** that determines the time locations of the maxima and minima of $s(t)$

eg: $\phi = 0 \implies s(0) = Acos(0) = A$, so $s(t)$ has a local maxima at 0
if $\phi ! = 0 \implies s(0) = Acos(\phi)$, which is not necessary A

Finding Maximum of the signal after t = 0

$$Acos(2\pi f_0 t + \phi) = A \equiv cos(2\pi f_0 t + \phi) = 1 \tag{3}$$

$$2\pi f_0 t + \phi = 2kpi \tag{4}$$

$$t = \frac{2kpi - \phi}{2\pi f_0},\ k \in \mathbb{Z} \tag{5}$$

f_0 **is the frequency in (Hz)**, determines the rate the signal oscillates

$$\text{Period } T_0 = \frac{1}{f_0} \tag{6}$$

1.2 Periodic Signals

A periodic signal persists for an infinity amount of time, and will always output the same waveform in any integer multiples of periods

$$s(t + T_0) = s(t + T_0),\ \text{for all t} \tag{7}$$

We can express ϕ as

$$\phi = 2k\pi + \phi_0, k \in F, \phi_0 \in [-\pi, \pi] \tag{8}$$

if $f_0 = 0$, for all values of $Acos(\phi)$ is a constant, hence a DC signal
We can ecpress any arbitrary sinusoidal signal as a linear combination of the two pure sinusoidal signals:

$$s(t) = R \times cos(2\pi f_0 t + \phi) \tag{9}$$
$$s(t) = A \times cos(s\pi f_0 t) + B \times sin(s\pi f_0 t) \tag{10}$$

where $A = Rcos(\phi)$ and $B = Rsin(\phi)$
Using the formulae:

$$sin(A \pm B) \equiv sin(A)cos(B) \pm cos(A)sin(B) \tag{11}$$
$$cos(A \pm B) \equiv cos(A)cos(B) \mp sin(A)sin(B) \tag{12}$$
$$tan(A \pm B) \equiv \frac{tan(A) \pm tan(B)}{1 \mp Atan(B)} \tag{13}$$
$$sin(3A) \equiv 3sin(A) - 4sin^3(A) \tag{14}$$
$$cos(3A) \equiv 4cos^3(A) - 3cos(A) \tag{15}$$
$$sin(P) + sin(Q) \equiv 2sin(\frac{1}{2}(P+Q))cos(\frac{1}{2}(P-Q)) \tag{16}$$
$$sin(P) - sin(Q) \equiv 2cos(\frac{1}{2}(P+Q))sin(\frac{1}{2}(P-Q)) \tag{17}$$
$$cos(P) + cos(Q) \equiv 2cos(\frac{1}{2}(P+Q))cos(\frac{1}{2}(P-Q)) \tag{18}$$
$$cos(P) - cos(Q) \equiv -2sin(\frac{1}{2}(P+Q))sin(\frac{1}{2}(P-Q)) \tag{19}$$
$$\tag{20}$$

1.3 Harmonics

Lets say we have Harmonic Frequencies $f_k, k \in \mathbb{Z}$

$$x_1(t) = cos(2\pi f_1 + \phi) \tag{21}$$
$$x_2(t) = cos(2\pi f_2 + \phi) \tag{22}$$
$$f_k = k \times f_0, k \in \mathbb{Z} \tag{23}$$

Proof Any linear combination of sinusoidal signals of harmonics of some frequency f_0 is a periodic signal

$$\text{For arbitrary } \{A_k\}\{\phi\} \tag{24}$$

$$s(t) = \sum_{k=1}^{N} A_k cos(2\pi f_0 t + \phi_k) \tag{25}$$

$$s(t) = \sum_{k=1}^{N} A_k cos(2\pi f_0 t + \phi_k) \tag{26}$$

$$s(t + T_0) = \sum_{k=1}^{N} A_k cos(2\pi f_0 t + 2\pi k + \phi_k) \tag{27}$$

$$\text{hence: } s(t + T_0) = s(t) \tag{28}$$

$$\tag{29}$$

1.4 Fourier Series

We can express any arbitrary periodic signal as a sum of harmonic sinusoids, let $x(t) = x(t + T_0)$ be a periodic signal with period T_0
Then there exist coefficients:

$$\{A_k\}_{k=0}^{\infty}, \{B_k\}_{k=0}^{\infty} \tag{30}$$

$$x(t) = \frac{A_0}{2} + \sum_{k=0}^{\infty} \left(A_k cos(2\pi f_0 kt) + B_k sin(2\pi f_0 kt)\right), \forall t \tag{31}$$

Calculating $\{A_k\}, \{B_k\}$ starting from $x(t)$ It can be shown that:

$$A_k = \frac{2}{T_0} \int_{-0.5T_0}^{0.5T_0} x(t) cos(2\pi kt) dt \tag{32}$$

$$B_k = \frac{2}{T_0} \int_{-0.5T_0}^{0.5T_0} x(t) sin(2\pi kt) dt \tag{33}$$

Fourier synthesis is the reverse process of starting from $\{A_k\}_{k=0}^{\infty}, \{B_k\}_{k=0}^{\infty}$ and generates a periodic signal

Equivalent Form:

$$s(t) = \frac{A_0}{2} \sum_{k=1}^{\infty} C_k cos(2\pi f_0 t + \phi_k) \tag{34}$$

where

$$C_k = \sqrt{A_k^2 + B_k^2} \; and \; \phi_k = tan^{-1}(\frac{B}{A}) \tag{35}$$

1.5 Spectrum

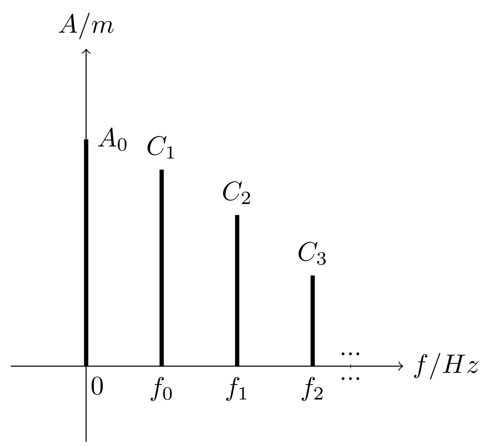
Fourier thyrm showed that all periodic signals can be written as an addictive linear combination of sinusoid signals
The spectrum of a periodic signal is the collection of amplitude, frequency and phase information that allow us to represent the signal as a linear combination

1. Time-domain: need knowledge of $f_0, A_0,$ and $\{x(t), t \in [-\frac{T_0}{2}, \frac{T_0}{2}]\}$
2. Frequency-domain: Would require $f_0, A_0, \{C_k\}_{k=1}^{\infty}, \{\phi_k\}_{k=1}^{\infty}$

The spectrum is given by the following collection:

Frequency	Amplitude	Phase
0	A_0	0
f_0	C_1	ϕ_1
$2f_0$	C_2	ϕ_2
$\vdots$	$\vdots$	$\vdots$
f_0	A_0	ϕ_1

1.5.1 Graphical plot of the spectrum



In which the vertical plots are the spectral lines

$$\begin{array}{c} \rightarrow \text{analysis} \rightarrow \\ \text{Time Domain} \Longleftrightarrow \text{Frequency Domain} \\ \leftarrow \text{syntheses} \leftarrow \end{array}$$

1.5.2 Benefits of spectrum

1. Often time-waveforms are very complicated while spectrun is more straight forward
2. We can compress data by only storing the important frequencies, shrinking file size (mp3)

3. Understanding the properties of the signal is often insightful on how to process item
    - Think of audio processing, mp3 is a format which removes all frequencies sampled beyond human's limits, also shrinking the file size
    - We can remove noise
4. Often easy to see how system affect a signal by determining what-happens to the signal spectrum as it is transmitted through the system
    - A radio receiver uses a susem called filler that filters out all frequencies in the received radio wave other than the frequeny of the channel we choosed

1.5.3 Fourier integral

For any periodic signal we might need to go beyond just harmonic signals:

$$x(t) = \int_0^{\infty} A(f)cos(2\pi ft) + B(f)sin(2\pi ft)df \tag{36}$$

where:

$$A(\omega) = \frac{1}{\pi}\int_{-\infty}^{\infty} x(t)cos(2\pi ft)dt \tag{37}$$

$$B(\omega) = \frac{1}{\pi}\int_{-\infty}^{\infty} x(t)cos(2\pi ft)dt \tag{38}$$

1.5.4 Limitations

1. Arbitrary continuous functions cannot be represented in practice and cannot be stored in computer
2. The involved integrals cannot be computed in general

1.6 Fourier Transform summary

Let $x(t) = x(t + T_0)$ be a periodic signal with period T_0, then there exist coefficients $\{A_k\}_{k=0}^{\infty}$, $\{B_k\}_{k=0}^{\infty}$ such that

$$
\begin{aligned}
x(t) &= \frac{A_0}{2} + \sum_{k=0}^{\infty} \left(A_k cos(2\pi f_0 kt) + B_k sin(2\pi f_0 kt) \right), \forall t && (39)\\
A_0 &= \frac{2}{T_0} \int_{-0.5T_0}^{0.5T_0} x(t) dt && (40)\\
A_k &= \frac{2}{T_0} \int_{-0.5T_0}^{0.5T_0} x(t) cos(2\pi f_0 kt) dt && (41)\\
B_k &= \frac{2}{T_0} \int_{-0.5T_0}^{0.5T_0} x(t) sin(2\pi f_0 kt) dt && (42)\\
&or\\
s(t) &= \frac{A_0}{2} \sum_{k=1}^{\infty} C_k cos(2\pi f_0 t + \phi_k) && (43)\\
&where\\
C_k &= \sqrt{A_k^2 + B_k^2}\ and\ \phi_k = tan^{-1}(\frac{B}{A}) && (44)
\end{aligned}
$$

A spectrum of a periodic signal is the collection of amplitude, frequency and phase information that allows us to represent the signal as a linear combination A_0 is the DC part of the signal Think of the integral as a Riemannian sum, that is, the average of signal over a period

1.7 Complex number involved in fourier transform

Complex number are an elegant way of expressing rotations, something periodic.

$$
\begin{aligned}
&Euler's\ Formula\\
&R(cos(\theta) + isin(\theta))\ (orRcis(\theta)) \equiv Re^{i\theta} && (45)
\end{aligned}
$$

Adding two complex number

$$
\begin{aligned}
z &= a + bi && (46)\\
z_1 &= a_1 + b_1 i && (47)\\
z + z_1 &= a + a_1 + b + b_1 i && (48)
\end{aligned}
$$

We can easily understand the concept of multiplication and division using Euler's formula

$$z = Re^{i\theta} \tag{49}$$
$$z_1 = R_1 e^{i\theta_1} \tag{50}$$
$$z \times z_1 = R \times R_1 \times e^{i(\theta+\theta_1)} \tag{51}$$

A conjugate of an imaginary number is an symmetric image of itself by the x-axis

$$z = a + bi \tag{52}$$
$$z* = a - bi \tag{53}$$
$$z \times z* = a^2 + b^2 \tag{54}$$
$$\Re(z) = \frac{z + z*}{2} \tag{55}$$
$$\Im(z) = \frac{z - z*}{2i} \tag{56}$$

hence

$$cos(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{2} \tag{57}$$
$$sin(\theta) = \frac{e^{i\theta} - e^{-i\theta}}{2i} \tag{58}$$

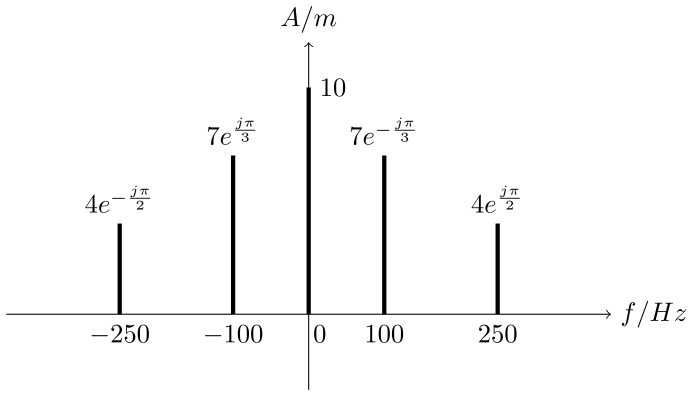
1.7.1 Complex Exponential Signal

We can use the conclusion above to express a signal with an complex exponential signal

$$z(t) = Ae^{2\pi f_0 t + \phi} \tag{59}$$

Where A is the magnitude and $\omega_0 + \phi$ is the angle linearly variating in time
We can plot an complex exponential signal spectrum for any given signal by seperating them into real part and imaginary part

$$let\ x(t) = 10 + 14cos(200\pi t - \frac{\pi}{3}) + 8cos(500\pi t + \pi/2) \tag{60}$$
$$\begin{aligned} x(t) = 10 + 7e^{-\frac{j\pi}{3}} e^{2j\pi(100)t} + 7e^{\frac{j\pi}{3}} e^{-2j\pi(100)t} \\ + 4e^{\frac{j\pi}{2}} e^{2j\pi(250)t} + 4e^{-\frac{j\pi}{2}} e^{2j\pi(250)t} \end{aligned} \tag{61}$$



Multiplying something by i jas an effect of rotating that something by 90 degrees. Using euler's formula, we can express any unit-length complex number as $e^{(2\pi it)}$ Where t is a real number, if t represents time, and plot the location of the complexnumber at the given time t. What we will see is a circle being drawn with a frequency of one cycle per second(Hz). We could adjust the speed of the rotation by multiplying a frequency f to the power, which gives us $e^{(2\pi ift)}$

Can it have negative frequencies? negative frequencies seems pretty abstract at first, but once you start to understand the idea of drawing a circle in the anti-clockwise direction, you will soon realise that having negative frequencies simply implies rotating in the clockwise direction.

Multipying the circle by a function $f(t)$ would then suggest wrapping the function around the circle, By taking the mean, finding the center of mass of the wrapped signal, at different time t with an integral, we can plot a diagram of the spectrum. When it all lined up, there will be an obvious peak. where the center of mass is shiped the farthest away from the signal.

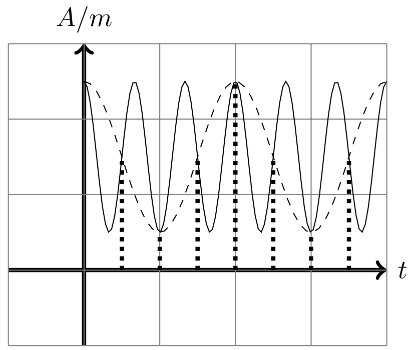
So the complex version of fourier integrals is:

$$x_k = \frac{1}{T_0} \int_0^{T_0} x(t) e^{-j2\pi f_0 kt} \tag{62}$$

2 Discrete Signals

2.1 Aliasing

Aliasing in the context of signal processing is simply the same name of two signals. When sampling a signal at a certain frequency, multiple signals at different frequency can produce the same signal For, example: $\hat{\omega} + 2\pi l$ and $2\pi l - \hat{\omega}$ are aliases of $\hat{\omega}$ as they are different discrete times signals defining the same signal values



As you can see from the diagram above, the two signal, both sampled at 2 Hz, produced the same value each sample, but are fundamentally, two different frequencies

The principle alias is the frequency in $[0, \pi]$

$$\hat{\omega} \triangleq \omega_0 T_s = \frac{\omega_0}{f_s} \tag{63}$$

Fix a discrete sinusioid of discrete-time frequency $\hat{\omega}$, then the frequency f_0 of the original continuous time sinusoid could have been any of the following:

$$f_0 = (\frac{\hat{\omega}}{2\pi})f_S \text{ or } f_0 = (\frac{\hat{\omega}}{2\pi})f_S + lf_s \text{ or } f_0 = lf_s - (\frac{\hat{\omega}}{2\pi})f_s \; wherel \in \mathbb{Z} \tag{64}$$

We cannot know which one of them is the true signal, unless we can fix more conditions

2.1.1 Sampling sinusoids

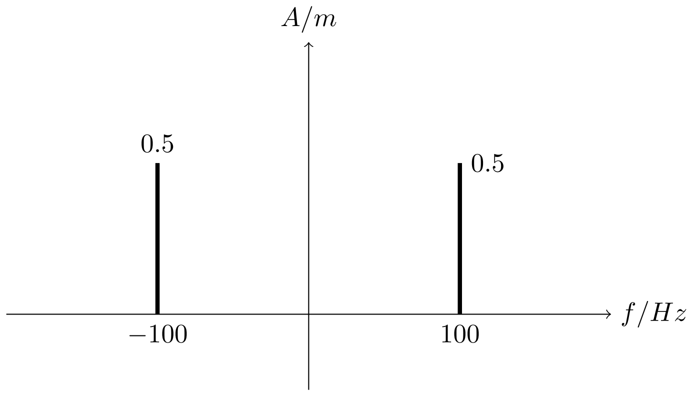
Take a sample of $x[n] = cos(0.4\pi n)$, and is equivalent to $x_2[n] = cos(2.4\pi n)$. An alias corresponding to the principal alias , by convention, is used. Once the name is fixed and f_s is known, we canguarantee that the samples came from

$$x[n] = \; cos(0.4\pi n) \text{ since } \hat{\omega} \in [0, \pi] \tag{65}$$

$$\implies \omega_0 = \hat{\omega} f_s \in [0, \pi \times f_s] \tag{66}$$

$$\implies f_0 = \frac{f_0}{2\pi} \in [0, \frac{f_s}{2}] \tag{67}$$

If the frequency of the original continuous time signal does not satisfy the condition mentioned above, then the resulting samples are identical to those obtained by sampling a lower frequency signal. So When $f_0 \notin [0, \frac{f_s}{2}]$, aliasing occurs.
Example:
Continuous-time signal $x(t) = cos(2\pi(100)t)$ with sampling frequency $f_s = 500Hz$.



The signal is baud-limited where $f_{max} = 100Hz$. The sampling frequency is larger than the Nyquist rate($2f_{max} = 200Hz$), so this is an over-sampling

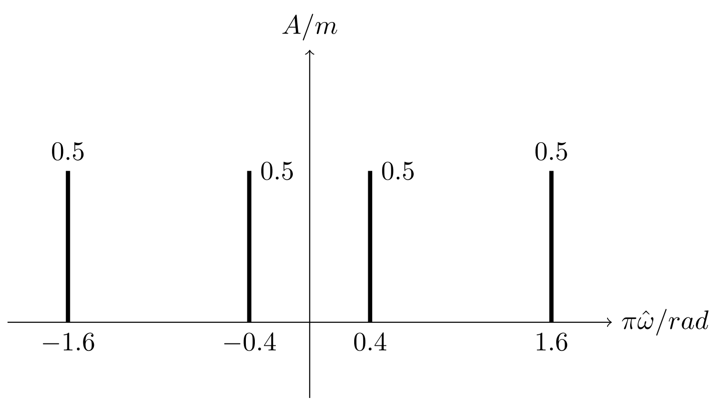
$$\hat{\omega} = \frac{2\pi \times 100}{f_s} = 0.4\pi \tag{68}$$

$$\hat{\omega} = 0.4\pi + 2\pi l \tag{69}$$

$$\hat{\omega} = 2\pi l - 0.4\pi \tag{70}$$

$$\text{where} l \in \mathbb{Z}$$

Hence the discrete-time spectrum plot looks like:



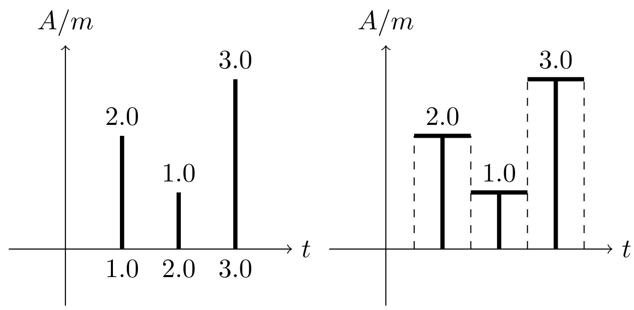
2.2 Discrete to Continuous Converter

Disrete to Continuous Converter (D2C for short), interpolates a smooth Continuous function through the given discrete samplesy[n]. A special case is $y[n] = Acos(\hat{\omega}n + \phi)$ In ideal case, $y(t) = Acos(2\pi f_0 t + \phi)$ should be produced. Only f_0 is required which can be found easily by $\hat{\omega}$.

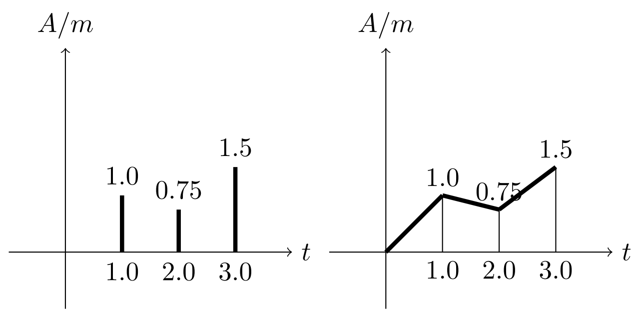
For more general signal, according to sampling theorem, we can still produce $y(t)$ as long as $f_s > 2f_{max}$
A popular model is

$s(t) = \sum_{n\in\mathbb{Z}} s[n]p(t - nT_s) where p(t) is a characteristic pulse of the converter$

Weighting signal as linear combination of time-shifted pulses.



If by linear interpolation



Ideal Pulse:

$$p(t) = sin(\frac{t}{T_s}) = \frac{sin(\frac{nt}{T_s})}{\frac{nt}{T_s}} \; where \; tin \pm \infty \tag{71}$$

$$p(n \times T_s) = \frac{sin(\pi n)}{\pi n} = \begin{cases} 1, & n = 0 \\ 0, & n = \pm 1, \pm 2, \dots \end{cases} \tag{72}$$

Pulse has infinite length $\Rightarrow$ All samples are needed to reconstruct a signal exactly from its samples.

Let $x[n] = cos(\hat{\omega}n + \phi)$, find $x_{out}(t)$ for the given sampling rates f_s

$$f_s = 500Hz,\ \hat{\omega} = 0.4\pi \tag{73}$$
$$\hat{\omega} \in [0, \pi] \Rightarrow \hat{\omega_p} = \hat{\omega} = 0.4\pi \tag{74}$$
$$\omega_{out} = \hat{\omega} \times f_s = 0.4\pi \times 500 = 200\pi \Rightarrow f_{out} = 100Hz \tag{75}$$

$$f_s = 500Hz,\ \hat{\omega} = 1.6\pi \tag{76}$$
$$\hat{\omega} \notin [0, \pi] \Rightarrow \hat{\omega_p} = 2\pi - \hat{\omega} = 0.4\pi \tag{77}$$
$$\omega_{out} = \hat{\omega} \times f_s = 0.4\pi \times 500 = 200\pi \Rightarrow f_{out} = 100Hz \tag{78}$$

$$f_s = 80Hz,\ \hat{\omega} = 2.5\pi \tag{79}$$
$$\hat{\omega} \notin [0, \pi] \Rightarrow \hat{\omega_p} = 2\pi\hat{\omega} = 0.5\pi \tag{80}$$
$$\omega_{out} = \hat{\omega} \times f_s = 0.5\pi \times 80 = 40\pi \Rightarrow f_{out} = 20Hz \tag{81}$$
$$\tag{82}$$

3 Brief introduction to Systems

3.1 Filters

Generally a filter can be classified as Running-Average or FIR(Finite impulse response).
A Filter is a system designed to remove source components or modify some characteristic of a signal.

A discrete-time system is a computational process for transforming one sequence in to another sequence (input to output)

Systems are often depicted as block diagram

$$x[n] \longrightarrow \boxed{\begin{array}{c} DT\ system \\ T\{\ \} \end{array}} \longrightarrow y[n]$$

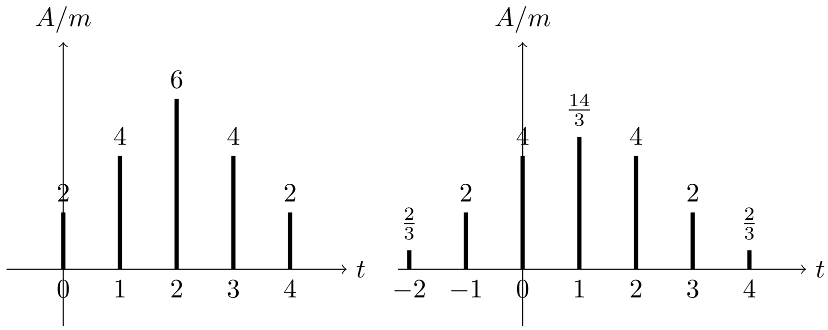
We seen D2C and C2D before, here we focus on systems where both input and output are discrete, such systems can be implemented with digital computations.

Mathematical Representation:
We represent the operation of a system by a operator $T\{\ \}$

$$y[n] = T\{x[n]\}\ \text{ such as:} \tag{83}$$
$$y[n] = (x[n])^2 \tag{84}$$
$$y[n] = \frac{\sum_{i=0}^{i=2} \{x[n-i]\}}{2} \tag{85}$$

Running-Average Filters (RAF):

Infact (85) mentioned above, is an 3-point-RA system Consider a finite length signal $x[n]$ where $x[n]$ itself is defined only in 0 4 Apply a 3-point-RA system, $y[n] = \frac{\sum_{i=0}^{i=2}\{x[n+i]\}}{3}$ to it



The output still has finite support and is longer, and it starts before the input starts. It uses the future values so it is a non-causal system
Causal means that the filter uses only the present and past values of the input. Non-causals are not implemented in real-time as future signal is not possible to obtain, but can be used in stored data.

$$y[n] = \frac{\sum_{i=0}^{i=2}\{x[nii]\}}{3} \; or \tag{86}$$

$$y[n] = \frac{\sum_{l=n-2}^{n}\{x[n]\}}{3} \; or \tag{87}$$

$$y[n] = y[n-1] + 1/3(x[n] - x[n-3]) \tag{88}$$

would be an causal RAF
If the length of x[n] is N, the length of the output of RAF after running through an M point RAF would be $N + M - 1$
Take an uncorrupted signal vector s, random noise z^i and a measurement result x^i, $x^i = s + z^i$, the ensemble average after k measurements is given by:

$$x_{ave}[n] = \frac{1}{k}\sum_{i=0}^{k} x^i[n] \tag{89}$$

$$= s[n] + \frac{1}{k}\sum_{i=0}^{k} z^i[n] \tag{90}$$

The $\frac{1}{k}\sum_{i=0}^{k} z^i[n]$ part is usually very small for random noise. The ensemble average gives a good estimate of the uncorrupte data over a large number of measurements k. If the data being measured cannot be repeated, then simply averaging a set of consecutive measurements often provides a reasonable estimate of the uncorrupted data.

An FIR filter is similar, but for each there's a certain coefficient determined by the position. $y[n] = \frac{1}{k}\sum_{l=0}^{M} h_l \times x[n-l]$, where h_l is one of the number in array $\{h_0, h_1, h_2, \ldots, h_M\}$ this takes no future values, so it is a causal system. If we take the array as another signal, we can have the relationship $y[n] = \frac{1}{k}\sum_{l=0}^{M} h[l] \times x[n-l]$, which is called convolution sum.

3.1.1 Linearity

If a system is linear, then if splitting the input into linear combination of two other functions and run them through the system seperately. We should get the same result as running the original system through it. So $y[n] = |x[n]|$ is obviously not an linear system.

Mathematically, we denote the process as this:

$$Letx[n] = \alpha x_1[n] + \beta x_2[n] T\{\alpha x_1[n] + \beta x_2[n]\} \quad = \alpha T\{x_1[n]\} + \beta T\{x_2[n]\} \tag{91}$$

3.1.2 Time-Invariant

If a input is delayed and as a response the output is delayed by the same amount, We said that it is a Time-Invariant system. Take Delayed input response $y_1[n]$ and Delayed-Output-Response$y[n-\phi]$

$$Let\ y[n] = \alpha x[n] \tag{92}$$
$$y_1[n] = \alpha x[n-\phi] \tag{93}$$
$$y[n-\phi] = \alpha x[n-\phi] \tag{94}$$
$$y_1[n] \equiv y[n-\phi] \tag{95}$$

So the system is Time-Invariant.

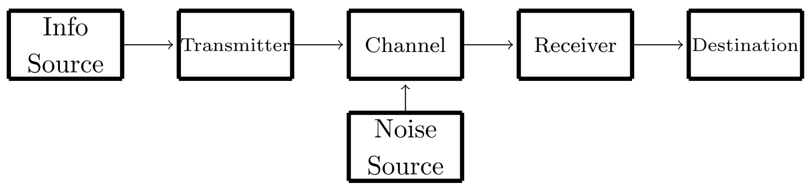
$$Let\ y[n] = n \times x[n] \tag{96}$$
$$y_1[n] = (n) \times x[n-\phi] \tag{97}$$
$$y[n-\phi] = (n-\phi) \times x[n-\phi] \tag{98}$$
$$y_1[n] \not\equiv y[n-\phi] \tag{99}$$

So the system is not Time-Invariant (Time-Variant)

Turns out, Impulse response is a complete characteristic of any Linear Time Invariant System

There used to be many form of communications: such as telegraphy, video and so on. They are all developed independently, with different methods of communication. As the network grew bigger they need a way to combine so they can work together. This increased focus on underlying principles of communication systems. Bell laboratories at AT&T back in the 1910s to 50s were the center of the most important developments. In 1948 Claude Shannon published his work "A mathematical theory of communication", which in a way created the Information Theory.

Shannon introduced a model that didn't depoend on a particular system or technology, As shown in the diagram below.



Now clear questions can be raised about any syetem.

1. How much information does a source produce?
2. What is the best way to encode a message in the transmitter?
3. Is there a limit to the amount of info we can send over the Channel
4. How badly noise affects the transmission?
5. How to model different communication sourses?
6. How to measure information?

Source might be discrete(letters, binary numbers) or analog(voltage, amplitude). All communication sources can be represented by binary sequences. The output of the source, instead of waiting for a certain input, is modeled as random. The idea of applying an ADC was revolutionary in 1948.

One of the simplest information source is just tossing a coin with two outcome with equal possibilities. We can use 0 and 1 to record the information, so only one bit is needed. If the coin is unfair and we can only gain 0, no bit (0 bit) is needed.

According to shannom, information is a measure of uncertainty. Hence the information of a source is a measurement of the average uncertainty in the outcome of the source. If we have a source of output with 256 possible symbol, how many bits do we need to send one?

We could use 256 bits which is 32 byte of data, which in today's standard is pretty small. But it is a gigantic amount of data back then (even for some systems today it is still pretty big). Since $log_2(256) = 8$, all we need is 1 byte(8 bit) of data.

Suppose we have a horse race with 8 houses each with integer number $n \in [0, 7]$, with winning chances

$$P(N = n) = \begin{cases} \frac{1}{2^{(n+1)}} n \in [0, 3] \\ \frac{1}{2} n >= 4 \end{cases} \tag{100}$$

How many bits do we need? We could use 1 byte of data, or we could use 3 bits of data if we need to save some space. Can we do even better then this? Well, if we model the length of the bits we used like another model, and find the expected length ($E(l)$) of data, we might be able to do even better.

$$Length(data) = \begin{cases} n + 1,\ n \in [0, 3] \\ 6,\ n >= 4 \end{cases} \tag{101}$$

$$E(l) = \sum_{i=0}^{3} \left\{ \frac{1}{2(i+1)} \times (i + 1) \right\} + \frac{1}{64} \times 6 \tag{102}$$

$$= 2 \tag{103}$$

Average bit length = 2, so we should be able to use 2 bits to model the entire thing, which should be enough. The way we calculated the expected length, is called the entropy of the source.

$$H = -\sum_{i=1}^{N} (P(i) \times log_2(P(i))) \tag{104}$$

$$\text{Entropy of free coin } P1 = P2 = 0.5 \tag{105}$$

$$H = -0.5\ log_2(0.5) + -0.5\ log_2(0.5) \tag{106}$$

$$= (-0.5 \times -1) + (-0.5 \times -1) \tag{107}$$

$$= 0.5 + 0.5 \tag{108}$$

$$= 1 \tag{109}$$

In an informal way, Entropy is a lower bound of the average number of bits required to represent the random variable.

4 Review

4.1 Signals

Signals are patterns of variation that represent or encode information.

1. They are mathematically represented as a function of one or more variables
2. Continuous Variable $\rightarrow$ Continuous Signals
   Discrete Variable $\rightarrow$ Discrete Signals

Time Representation

$$\begin{array}{ccc}
\text{Continuous-Time Signals} & \text{Sampling} & \text{Discrete-Time Signals} \\
x(t) & \underset{x[n]=x(n\times T_s)}{\longrightarrow} \quad \longleftarrow & x[n] \\
\text{function: } \mathbb{R}\rightarrow\mathbb{R} & \text{Interpolation} & \text{sequence*[1]: } \mathbb{Z}\rightarrow\mathbb{R} \\
 & x(t)=\sum_n x[n]p(t-nT_s) &
\end{array}$$

Fourier Analysis $\downarrow$ $\uparrow$ Fourier Syntheses

Fourier Transform $\uparrow$ $\downarrow$ Inverse Fourier Transform

Line Spectrum (Periodic Signals)
$\{(f_k = kf_0, X_k)\}_{k\in\mathbb{Z}}$
Fourier Transform*[2]:
$X(f)\ f \in \mathbb{R}$
(finite-energy signals)

Discrete-Time FT $X(\hat{\omega}),\ \hat{\omega} \in [-n, n]$
Discrete FT (DFT):
$X(\frac{2\pi k}{N}),\ k \in [0, N-1]$
N is signal length
Fast Algorithm FFT $\begin{cases} Fast \\ Fourier \\ Transform \end{cases}$

Frequency Representation

[1] Digital Signals only take finite number of values, we ignore qualization in this course
[2] We did not cover the continuous-time Fourier Transformation, it will be introduced at a high-level in the future as a generalization of the line-spectrum for non-periodic signals

4.2 Fourier Series

Any periodic real signals (which can be proven by) $x(t) = x(t + \frac{1}{f_0})$ can be written as a (possibly infinite) amount of addictive linear combination of (complex) sinusoidal signals:

$$x(t) = X_0 + \sum_{k=1}^{+\infty} X_k e^{2\pi i \times f_0 kt} + X_k * e^{-2k\pi i \times f_0 t} \tag{110}$$

$$= \sum_{k=-\infty}^{\infty} X_k e^{2\pi i \times f_0 kt} (X_{-k} = X*_k) \tag{111}$$

$$\equiv x(t) = A_0 + \sum_{k=1}^{+\infty} A_k cos(2\pi f_0 kt + \phi_k) \tag{112}$$

$$\text{Where } X_0 = A_0 \text{ and } X_k = \frac{1}{2} A_k e^{i\Phi k}$$

4.2.1 Line Spectrum

The set of pairs: $\{\ldots,(-f_k,\frac{1}{2}X_k*),\ldots,(-f,\frac{1}{2}X_1*),(0,X_0),(f_1,\frac{1}{2}X_1),\ldots,(f_k,\frac{1}{2}X_k)\}$
(Why Fourier Representation?)

1. Can reveal information about the signal that is not easy to spot in the time-domain representation*[Refer to "Finding a note"]*
2. As such, it helps identify structure/properties of the signal
3. These properties can be used later as information that helps processing a signal
4. For example, cans use Fourier representation for denoising:
   - Idea is that noise usually has high-frequencies that are "separated" from the useful frequencies of the signal, so denoising can be done in frequency domain*[Refer to "Noise-Filtering-Solutions"]* in ThinkDSP
5. Can also use to compress a signal
   - Idea is only to use dominant frequencies (i.e. Frequencies with large magnitude) *[Recall the square wave approximation]*
6. The input-output relation of an LTI(Linear-Time-Invariant) system is simpler to view in Frequency domain as is a simple multiplication
   - Algorithmic gain can gain easier infuition about about filter operations
   - Computational gain: fast way to perform convolution thanks to numpy's FFT

Fourier Synthesis

$$\sum_{-\infty}^{\infty} X_k e^{(2\pi i f_0 kT)} \tag{113}$$

Fourier Analysis

$$X_k = \frac{1}{T_0}\int_{\frac{T_0}{2}}^{-\frac{T_0}{2}} x(t)e^{(-2\pi i f_0 kT)} \tag{114}$$

Complex Amplitude $X_k \leftarrow$ Complex Numbers

- Magnitude $|X_k|$
- Phase $arg(X_k)$
- Polar Form: $X_k = |X_k|e^(iarg(X_k))$

Recall Euler's Formula:

$$e^{i\theta} = cos(\theta) + isin(\theta) \tag{115}$$

$$cos(\theta) = \Re(z) = \frac{1}{2}(z + z*) = \frac{1}{2}(e^{i\theta} + e^{-i\theta}) \tag{116}$$

$$sin(\theta) = \Im(z) = \frac{1}{2i}(z - z*) = \frac{1}{2i}(e^{i\theta} - e^{-i\theta}) \tag{117}$$

4.3 Sampling

$x[n] = x(nT_s) = x[n\frac{1}{f_s}]$ ...of sinusoids

$$x(t) = cos(2\pi f_0 t + \Phi) \tag{118}$$

$$\Rightarrow x[n] = cos(\frac{\omega_0}{f_s} n + \Phi) \mid \omega_0 = 2\pi f_0 \tag{119}$$

$$\triangleq cos(\hat{\omega} n + \Phi) \mid \hat{\omega} \triangleq \frac{\omega_0}{f_s} = \omega_0 T_s \tag{120}$$

Where thelast one is normalized radian Discrete-Time Frequency

4.3.1 Aliasing

$$cos(\hat{\omega} n + \Phi) = cos((2\pi l \pm \hat{\omega}) n + \Phi) \tag{121}$$

Infinite number of discretie-time signals can define the same values

$$\hat{\omega},\ 2\pi l + \hat{\omega},\ 2\pi l - \hat{\omega} \text{ aliases} \tag{122}$$

So we defined that the principal alias is the frequency in [0,n]

4.3.2 Interpolation

By convention use the alias corresponding to the principal alias, principal alalias

$$\hat{\omega} \in [0, \pi] \Rightarrow \omega \in [0, \pi f_s] \Rightarrow f_0 \in [0, \frac{f_s}{2}] \tag{123}$$

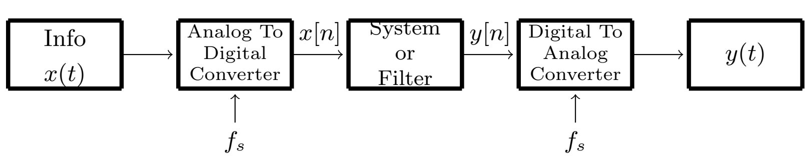
If original frequency not in $[0, \frac{f_s}{2}]$, then aliasing occurs. Recall the sampling theorem, $f_s \geq 2f_{max}$
A cut-time signal with frequencies no higher than f_{max} can be reconstructed exactly from its samples $x[n] = x(nT_s)$ provided that the sampling rate $f_s = 1/T_s$ satisfies $f_s > 2f_{max}$ where $2f_{max}$ is the Nyquist rate
How reconstruction works:

$$x(t) = \sum_{n \in \mathbb{Z}} x[n] p(t - nT_s) \tag{124}$$

Ideal pulse:

$$p(t) = sinc(\frac{t}{T_s}) = \frac{sin(\frac{nt}{T_s})}{\frac{nt}{T_s}} \tag{125}$$

Practical Pulses: Zero-order hold Linear Interpolation



4.4 Linear Time-Invariant Filters (LIT)

Fully characterized by their impulse-response $h[n]$ and the output to any input signal $x[n]$ is the convolution of the input with the impulse response

$$y[n] = h[n] * x[n] \tag{126}$$

Focus: FIR filters and finite-length inputsignals

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k] \tag{127}$$

Support set of the output: 0,1,. . . ,N+M-1

$$\sigma[n] \rightarrow \text{FIR} \rightarrow h[n]$$
$$x[n] \rightarrow \text{FIR} \rightarrow y[n] = x[n] * h[n]$$

Key Idea
This way, we can view filters as signals, specifically LIT systems as their impulse response. ⇒Can talk about frequency response of an LIT filter, i.e. Frequency representation of its impulse response

Fastway to perform convolution via FFT

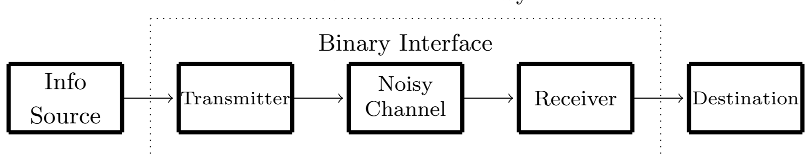
$$y[n] = h[n] * x[n] \iff Y[k] = H[K]X[K] \tag{128}$$

In the above equation, $Y[k]$, $H[K]$, and $X[K]$ are Discrete Fourier Transform coefficients.
Convolution in time domain is multiplication in frequency domain.

Examples, Use of moving-averate filter for:

1. Smoothening: get rid of oscillations to reveal a trend
2. Denoising: get rid of zero-mean random noise

Basics of Communication Systems



Fundamental Questions

1. Smoothening: get rid of oscillations to reveal a trend
2. Denoising: get rid of zero-mean random noise

Or can we achieve vanishing probs of error at a non trival(ie non zero) rate of communication
Question 1: How to measure information of a data source?
Model data sources as a stochastic process/signal. One of the simplest such model is the output of the data source is a discrete random variable X with probability mass function $P_x(x)$

Everyout is an independent relation of the Random variables x (next symbol is independent of previous variables)

[Shannon] Information of such a source is measured by the entropy of X:

$$H(x) = \sum_{x \in A_x} P(x) \times log_2(\frac{1}{2P(x)}) \tag{129}$$

Entropy is a measure of uncertainty about the outcome of the source.
How to compress information?

1. Use binary codes, it, to each possible source outcome assign a binary sequence (aka codeword).
2. The length of each codeword can depend on the probability of the corresponding symbol.
3. Goal is to design "Good" codes of minimum average length.
4. "Good" codes are instantaneous (Self-Punctuating) codes.
5. The limit of how small the minimum average length of an instanteous code can be , is entropy of the source

$$L(C) \geq H(X) \tag{130}$$

For Python Parts:

(a) If/Else statements

(b) For Loops

(c) functions

(d) Numpy Arrays