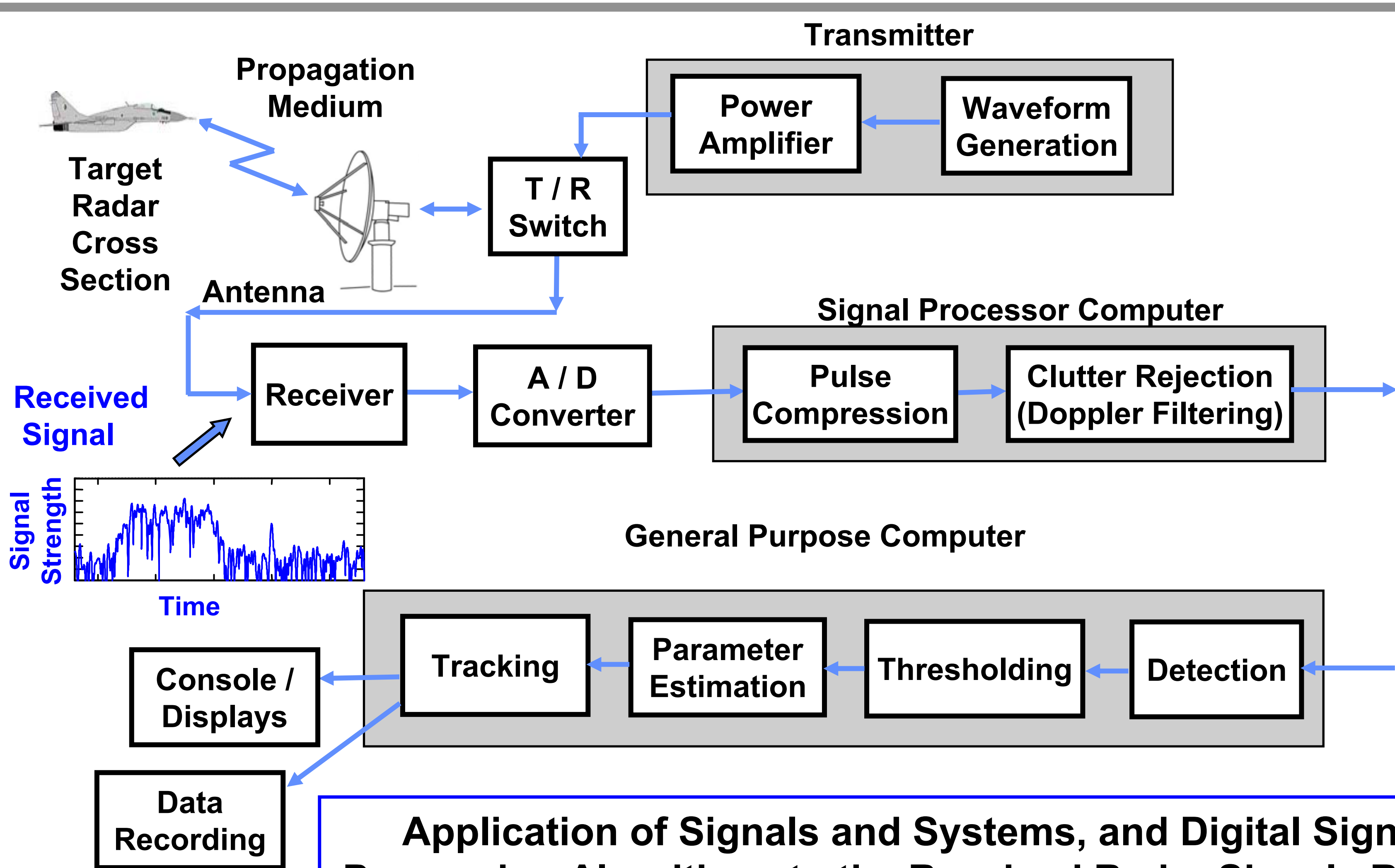
Radar Systems Engineering
Lecture 3
Review of Signals, Systems and Digital Signal Processing

Dr. Robert M. O'Donnell
IEEE New Hampshire Section
Guest Lecturer

Block Diagram of Radar System



Application of Signals and Systems, and Digital Signal Processing Algorithms to the Received Radar Signals Result in Optimum Target Detection

Reasons for Review Lecture

- Signals and systems, and digital signal processing are usually one semester advanced undergraduate courses for electrical engineering majors

- In no way will this 1+ hour lecture to justice to this large amount of material

- The lecture will present an overview of the material from these two courses that will be useful for understanding the overall Radar Systems Engineering course
  – Goal of lecture- Give non EE majors a quick view of material; they may wish to study in more depth to enhance their understanding of this course.

- UC Berkeley has an excellent, free, video Signals and Systems course (ECE 120) online at //webcast.berkeley.edu
  – http://webcast.berkeley.edu/course_details.php?seriesid=1906978405
  – Given in Spring 2007

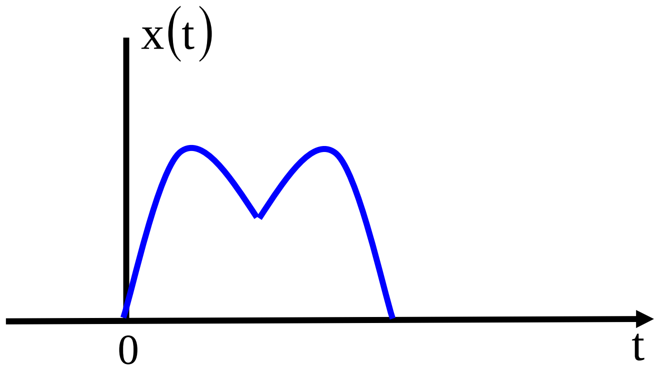
Signal Processing

- Signal processing is the manipulation, analysis and interpretation of signals.
- Signal processing includes:
  - Adaptive filtering / thresholding
  - Spectrum analysis
  - Pulse compression
  - Doppler filtering
  - Image enhancement
  - Adaptive antenna beam forming, and
  - A lot of other non-radar stuff (Image processing, speech processing, etc.
- It involves the collection, storage and transformation of data
  - Analog and digital signal processing
  - A lot of processing “horsepower” is usually required

Outline

- Continuous Signals
- Sampled Data and Discrete Time Systems
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

Continuous Time Signal



Examples:

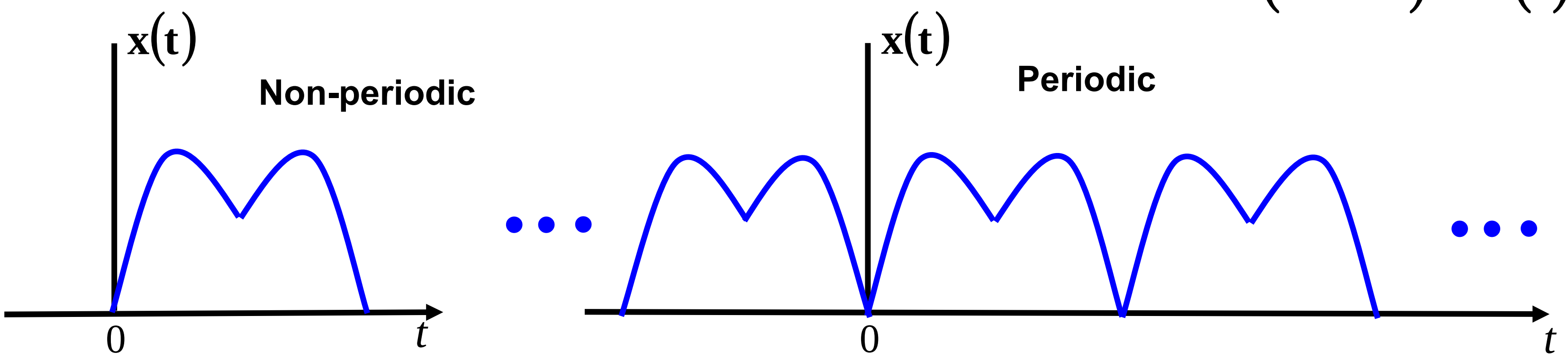
$$x(t) = 100\sin(\pi t) - 79\cos(3\pi t)$$

$$x(t) = 12t - 300$$

$$x(t) = t^2 - t^3 + 25t^{-5}$$

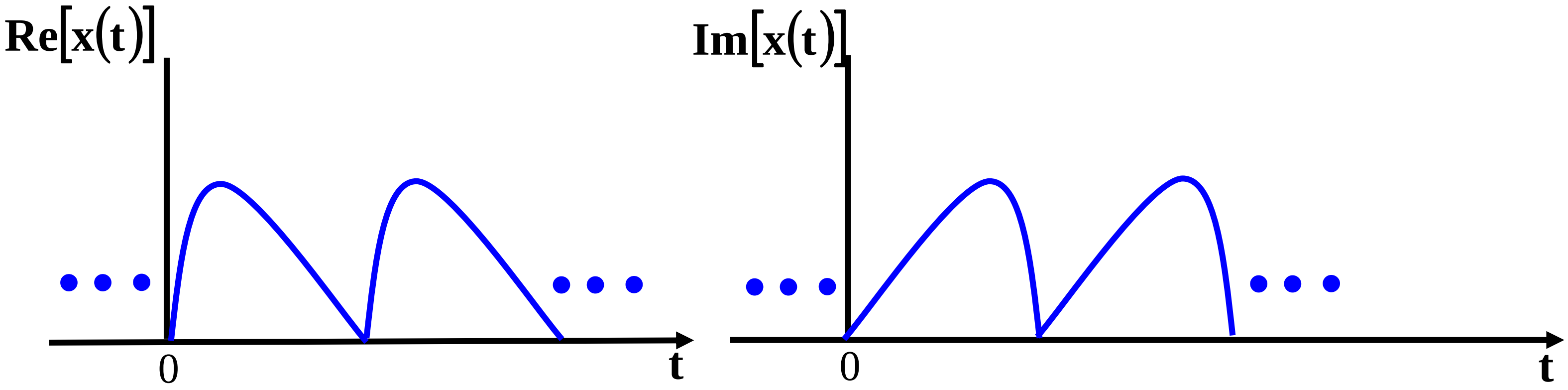
Continuous Time Signal

$$x(t+\Delta t) = x(t)$$



- Types of continuous time signals
  - Periodic or Non-periodic

Continuous Time Signal

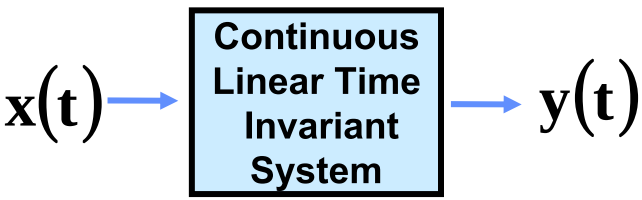


$x(t)$ **is a complex periodic signal**

- **Types of continuous time signals**
  - **Periodic or Non-periodic**
  - **Real or Complex**

    Radar signals are complex

Continuous, Linear, Time Invariant Systems



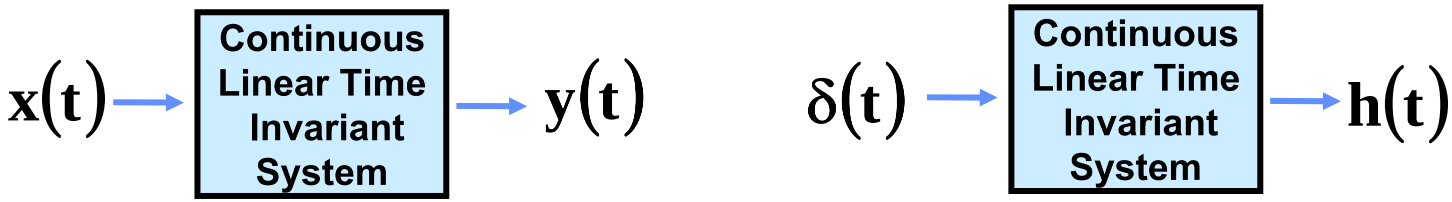
$$y(t) = T[x(t)] \qquad T = \text{Operator}$$

- **Continuous**
  - If $x(t)$ and $y(t)$ are continuous time functions, the system is a continuous time system

- **Linear**
  - If the system satisfies $T[\alpha x_1(t) + \beta x_2(t)] = \alpha y_1(t) + \beta y_2(t)$

- **Time Invariant**
  - If a time shift in the input causes the same time shift in the output

Linear Time Invariant Systems
(Delta Function)



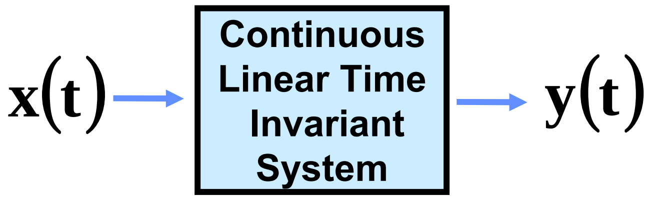
Properties of Delta Function

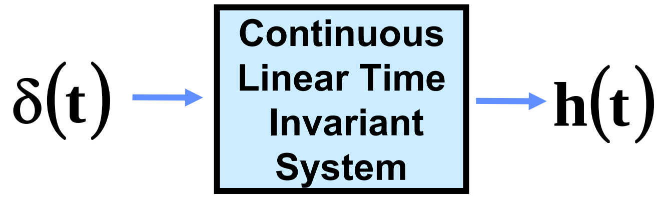
$$\delta(t) = \begin{cases} 0 & t \neq 0 \\ \infty & t = 0 \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1$$

- **The impulse response $h(t)$ is the response of the system when the input is $\delta(t)$**

Linear Time Invariant Systems



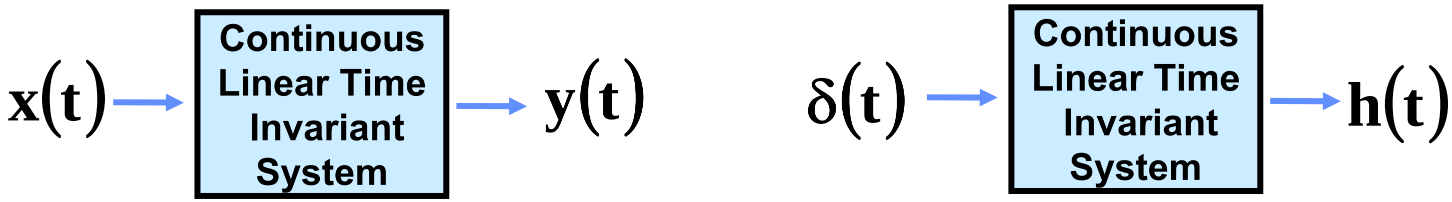


Definition : Convolution of Two Functions

$$x_1(t) * x_2(t) \equiv \int_{-\infty}^{\infty} x_1(\tau)\, x_2(t-\tau)\, d\tau$$

Reversed and Shifted

Linear Time Invariant Systems



Convolution of $x(t)$ and $h(t)$

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

- **The output of any continuous time, linear, time-invariant (LTI) system is the <u>convolution</u> of the input $x(t)$ with the impulse response of the system $h(t)$**

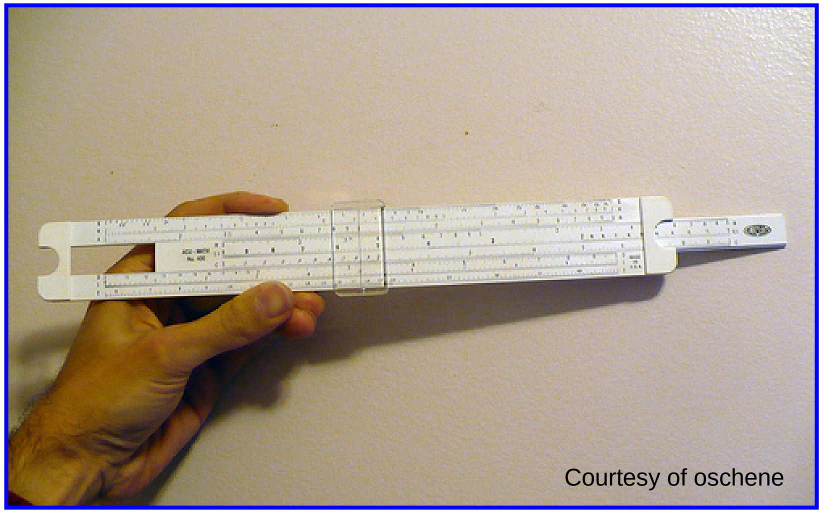
Why not Analog Sensors and Calculation Systems ?

Voltmeter

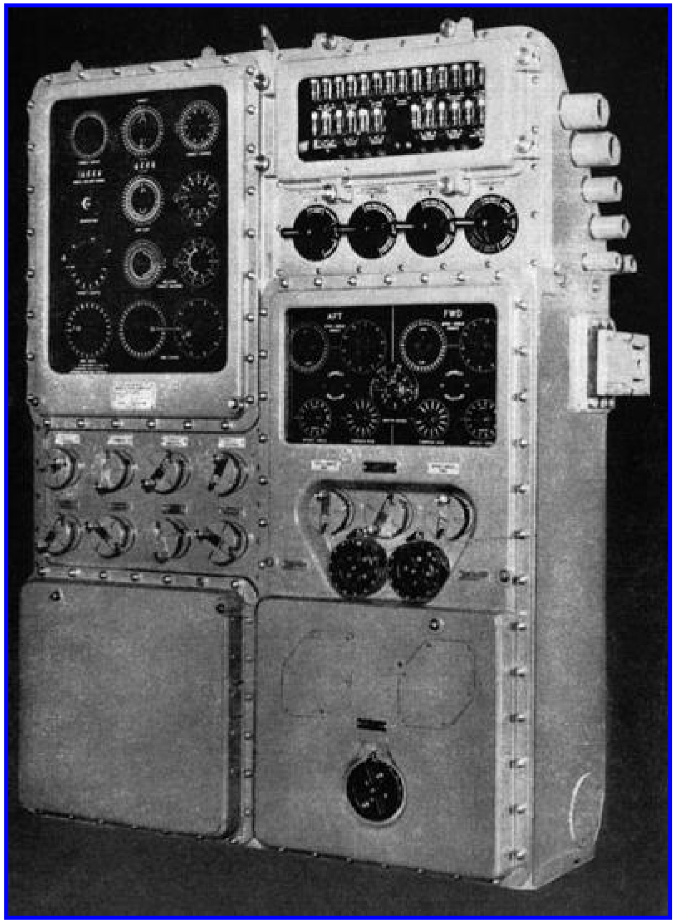
Courtesy of Hannes Grobe

Disadvantages

- **Measurement Repeatability**
- **Environmental Sensitivity**
- **Size**
- **Complexity**
- **Cost**


Courtesy of oschene

Slide Rule


Courtesy of US Navy

Torpedo Data Computer (1940s)

Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
  - General properties
  - A/D Conversion
  - Sampling Theorem and Aliasing
  - Convolution of Discrete Time Signals
  - Fourier Properties of Signals
    - Continuous vs. Discrete
    - Periodic vs. Aperiodic
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

Sampled Data Systems

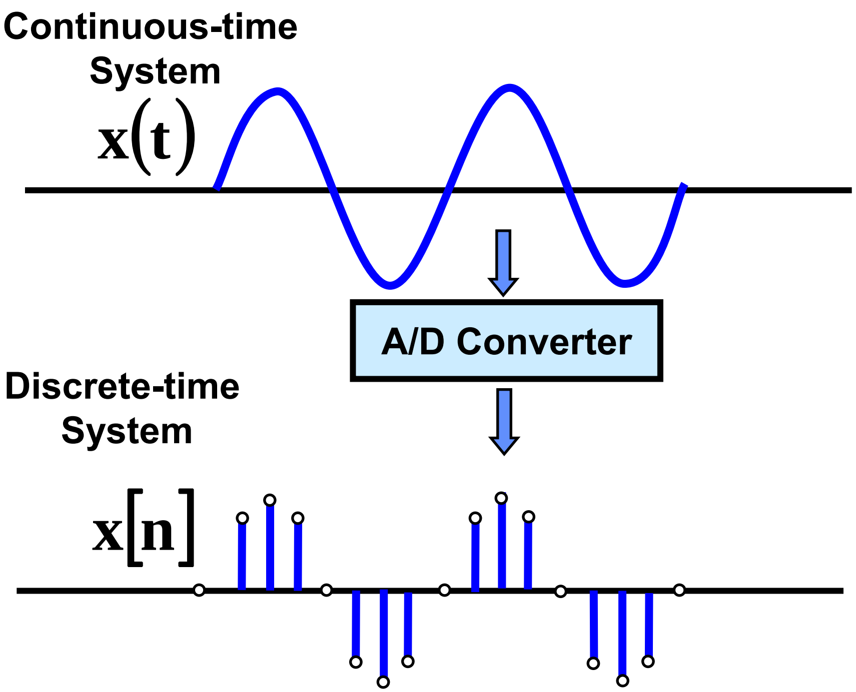
- **Digital signal processing deals with sampled data**

- **Digital processing differs from processing continuous (analog) signals**

- **Digital Samples are obtained with a “Sample and Hold” (S/H) Amplifier followed by an “Analog-to-Digital” (A/D) converter**
  - **Sampling rate**
  - **Word length**

Waveform Sampling

- Sampling converts a continuous signal into a sequence of numbers

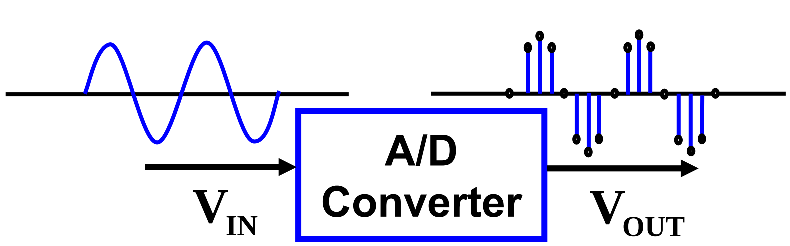


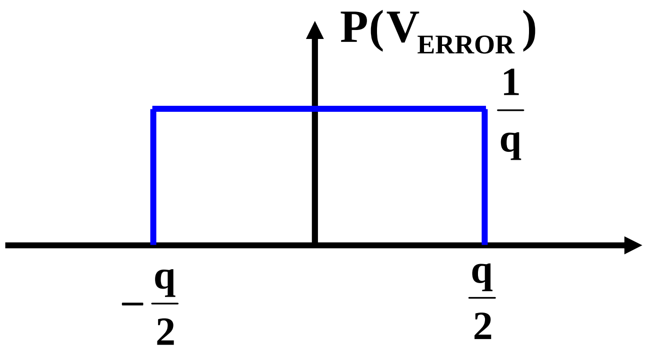
- Radar signals are complex

Outline

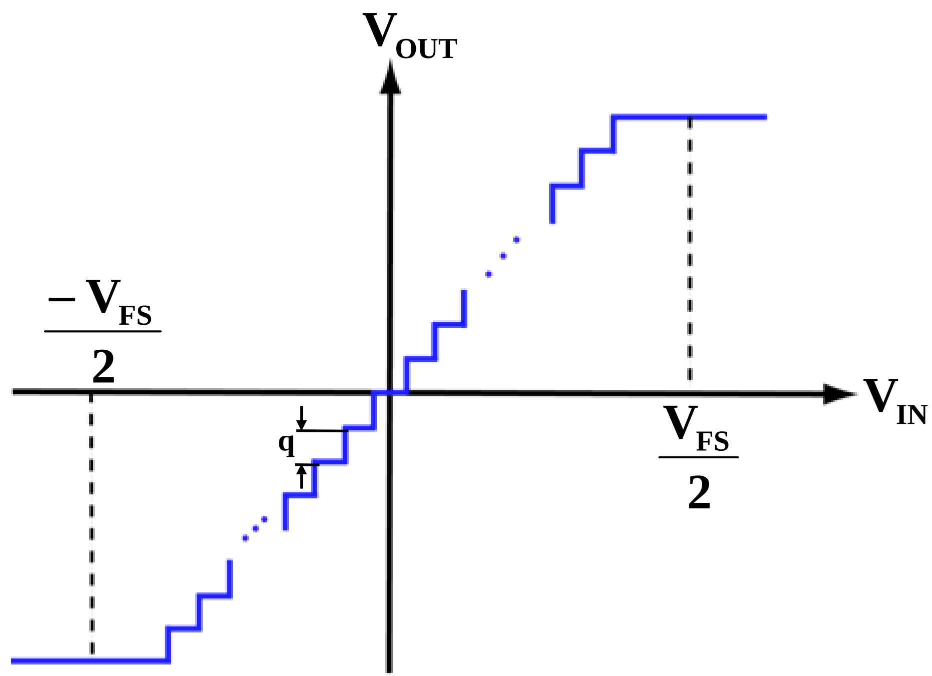
- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
  - General properties
  - A/D Conversion
  - Sampling Theorem and Aliasing
  - Convolution of Discrete Time Signals
  - Fourier Properties of Signals
    - Continuous vs. Discrete
    - Periodic vs. Aperiodic
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

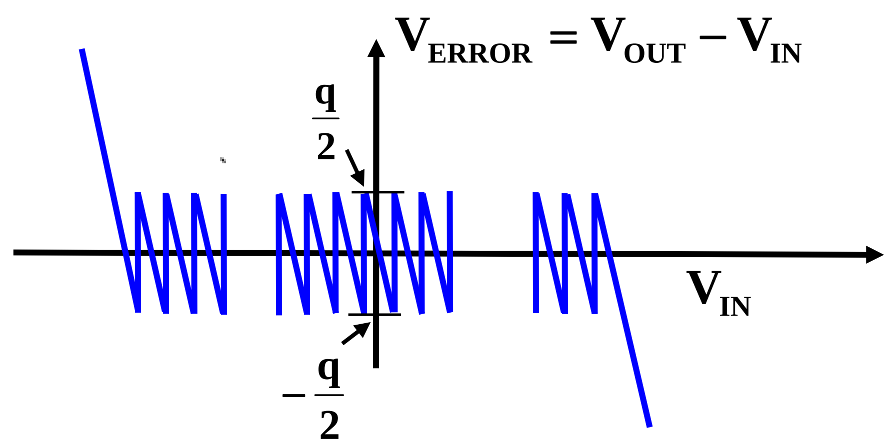
Ideal Analog to Digital (A/D) Converter



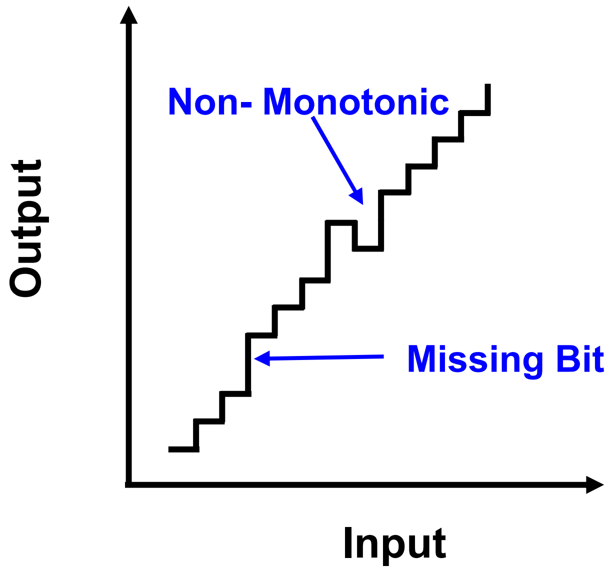


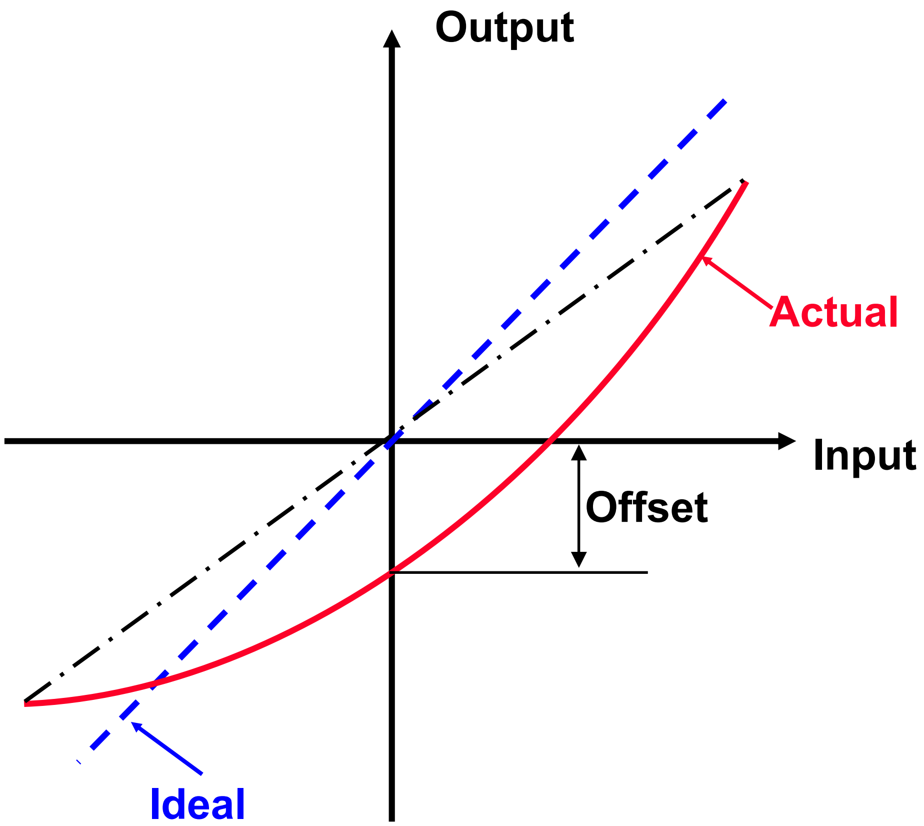
$$\sigma^2_{V_{ERROR}} = \frac{q}{12}$$





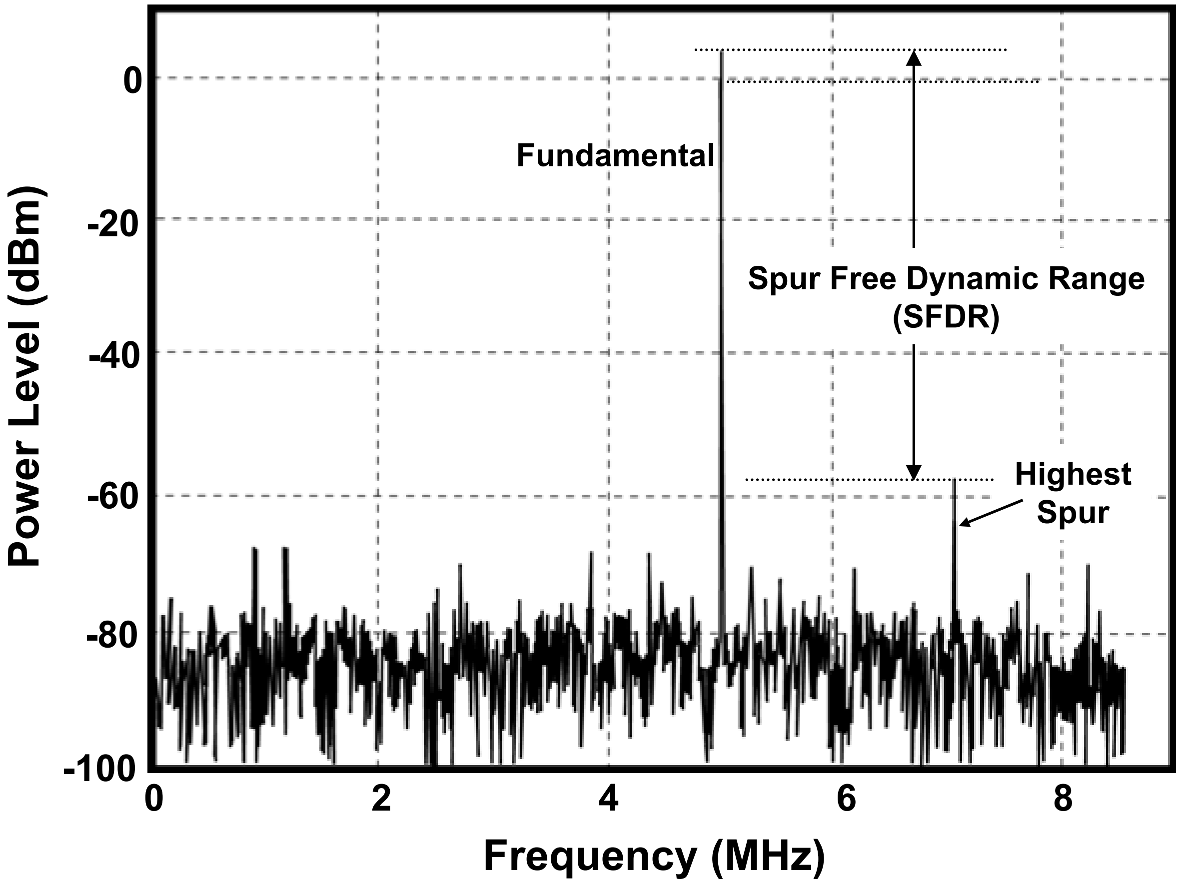
“Non-Perfect Nature” of A/D Converters





- Gain
- Missing bits
- Monotonicity
- Offset
- Nonlinearity
- Missing bits

Single Tone A/D Converter Testing



For Ideal A/D   S/N=6.02N + 1.76 dB

IEEE New Hampshire Section
IEEE AES Society

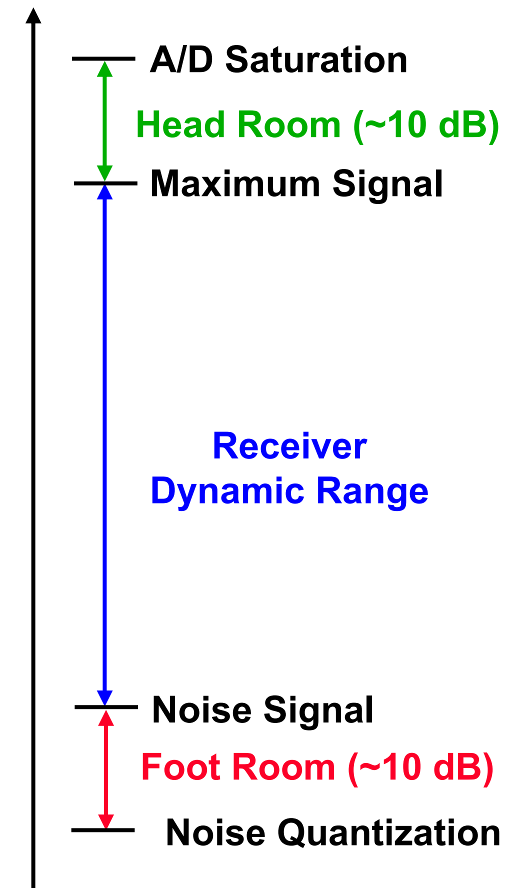
A/D Word Length

- A / D output is signed N bit integers
  - Twos complement arithmetic
  - Quantization noise power = $1/12$
- Signal-to-noise ratio $\left(\mathrm{SNR},\ S^2/N_o\right)$, must fit within the word length:
  - S^2= maximum signal power (target, jamming, clutter)
  - N_o= thermal noise power in A / D input

$$2^{L-1} > \alpha S \qquad\qquad 1/12 < N_o$$

  - Typically, $\alpha \approx 4$ to reduce clipping (limiting)
- Required word length: $L > \left(\mathrm{SNR}_{DB}/6\right) + 1.2$

$$\mathrm{SNR}_{DB} = 10\log_{10}\mathrm{SNR}$$

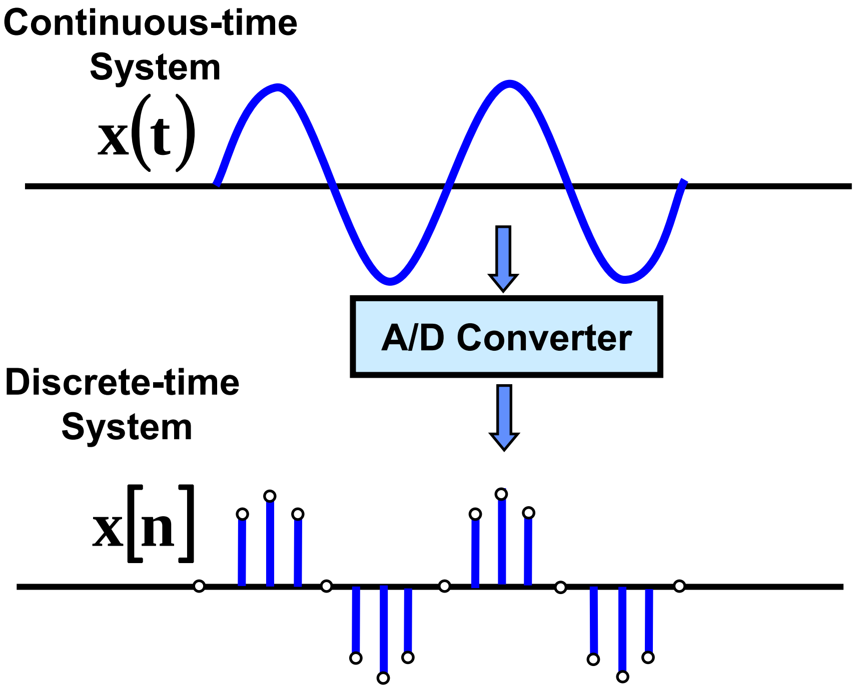


Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
  - General properties
  - A/D Conversion
  - Sampling Theorem and Aliasing
  - Convolution of Discrete Time Signals
  - Fourier Properties of Signals
    - Continuous vs. Discrete
    - Periodic vs. Aperiodic
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

Waveform Sampling

- Sampling converts a continuous signal into a sequence of numbers



- Radar signals are complex

Sampling - Overview

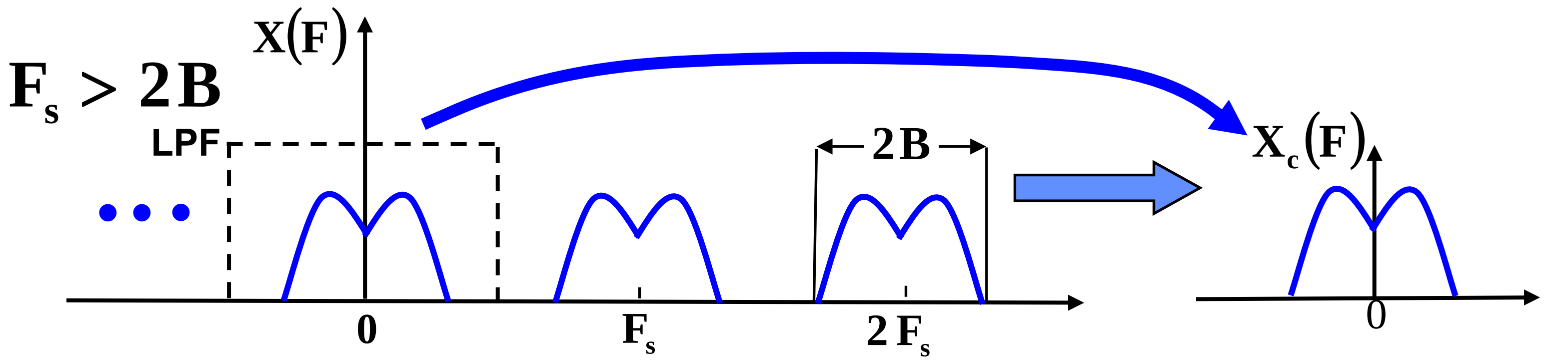
- **Sampling Theorem constraint (a.k.a. Nyquist criterion) to prevent "aliasing":**
  - **For continuous aperiodic signals:**

$$F_s \geq 2B \quad F_s = \text{Sampling Frequency}$$

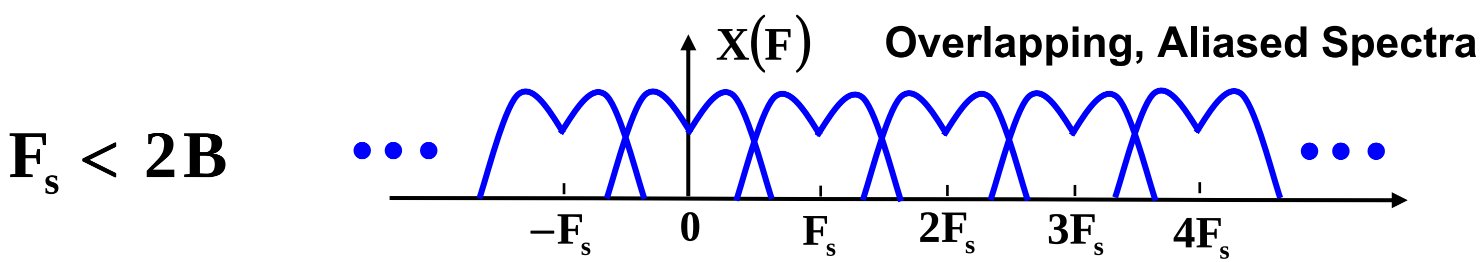
- **Nyquist criterion:**
  - **Permits reconstruction via a low pass filtering**
  - **Eliminates Aliasing**

Signal Sampling Issues

- **Signal Reconstruction**



- **Elimination of "Aliasing"**



The Sampling Theorem

- If $x_c(t)$ is strictly band limited,

$$X(F) = 0 \quad \text{for} \quad |F| > B$$

then, $x_c(t)$ may be uniquely recovered from its samples $x[n]$ if

$$F_S = \frac{2\pi}{T_S} \geq 2B$$

The frequency B is called the ***Nyquist frequency***, and the minimum sampling frequency, $F_S = 2B$, is called the ***Nyquist rate***

Spectrum of a Sampled Signal

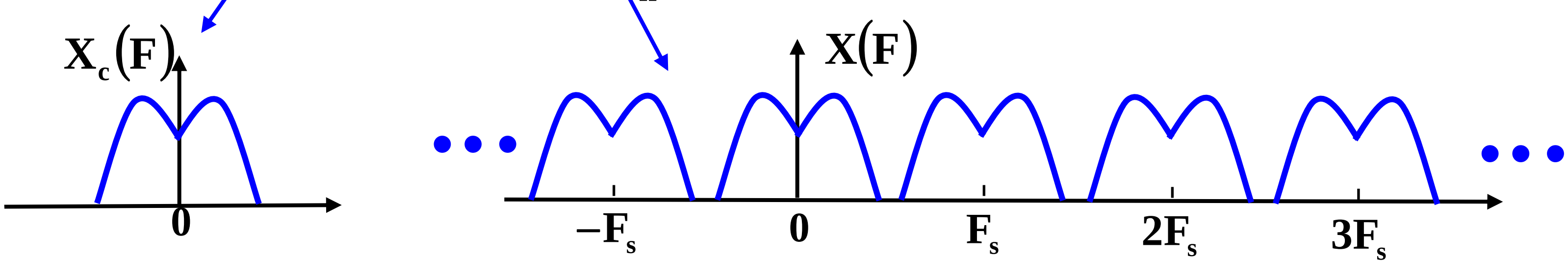
- **Sampling periodically replicates the spectrum**
  - **Fourier transform of a sampled signal is periodic**

- **If $X_c(F)$ and $X(F)$ are the spectra of $x_c(t)$ and $x[n]$**

$$X_c(F) = \int_{-\infty}^{\infty} x_c(t)\, e^{-j2\pi Ft}\, dt$$

$$X(F) = \int_{-\infty}^{\infty} \left(\sum_{n=-\infty}^{\infty} g(t)\, \delta(t - nT) \right) e^{-j2\pi Ft}\, dt$$

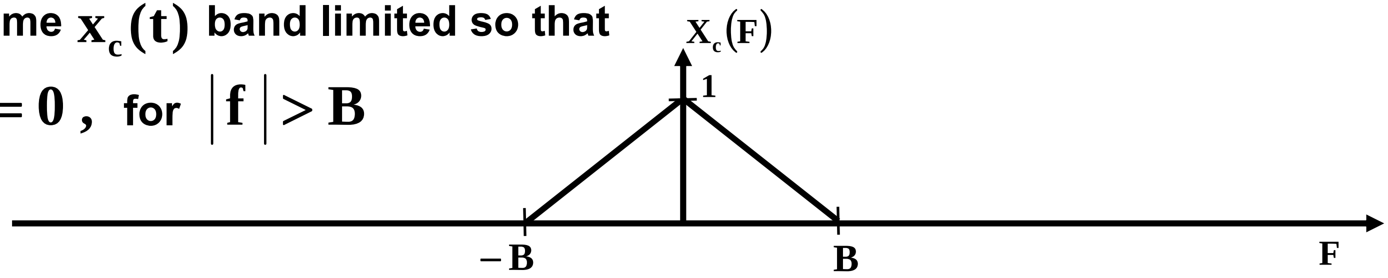
$$= \sum_{n=-\infty}^{\infty} x[n]\, e^{-j2\pi nF/F_s}$$



Distortion of a Signal Spectrum by "Aliasing"

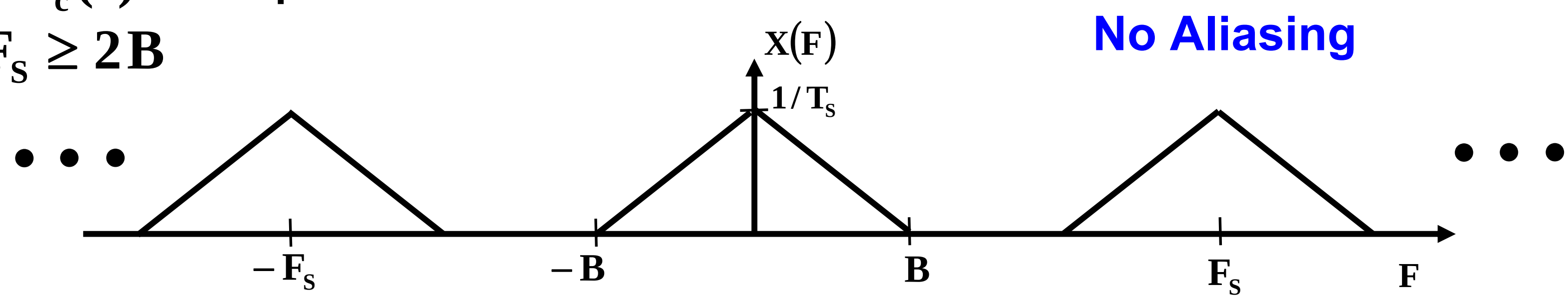
- Assume $x_c(t)$ band limited so that

  $X(f) = 0\ ,\ \ \text{for}\ \left|f\right| > B$



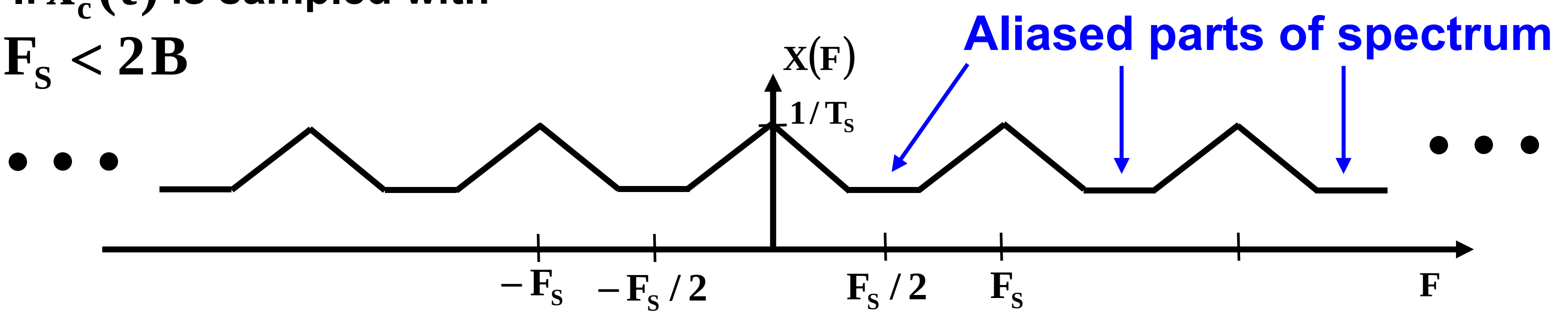
- If $x_c(t)$ is sampled with

  $F_S \geq 2B$

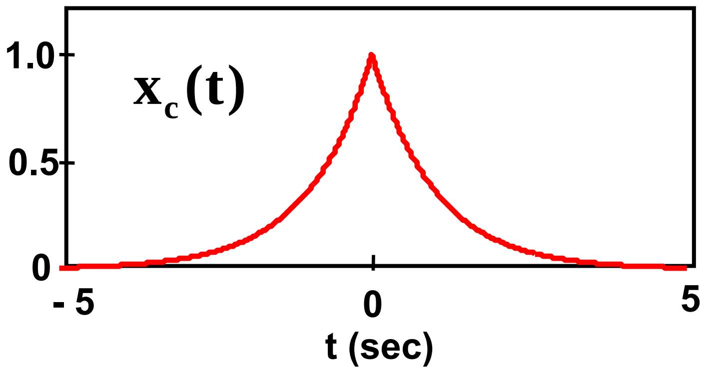


- If $x_c(t)$ is sampled with

  $F_S < 2B$



Effect of Sampling Rate on Frequency



Continuous Signal

$$x_c(t) = e^{-A|t|} \,,\; A > 0$$

Its Fourier Transform

$$X_c(F) = \frac{2A}{A^2 + (2\pi F)^2}$$

Sampled Signal

$$x[n] = x_c(nT) = e^{-AT|n|} = \left(e^{-AT}\right)^{|n|} = a^{|n|} \qquad a = e^{-AT} \,,\; F_S = \frac{1}{T}$$

0

Its Fourier Transform

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} = \frac{1-a^2}{1-2a\cos\omega + a^2} \quad , \quad \omega = 2\pi \frac{F}{F_S}$$

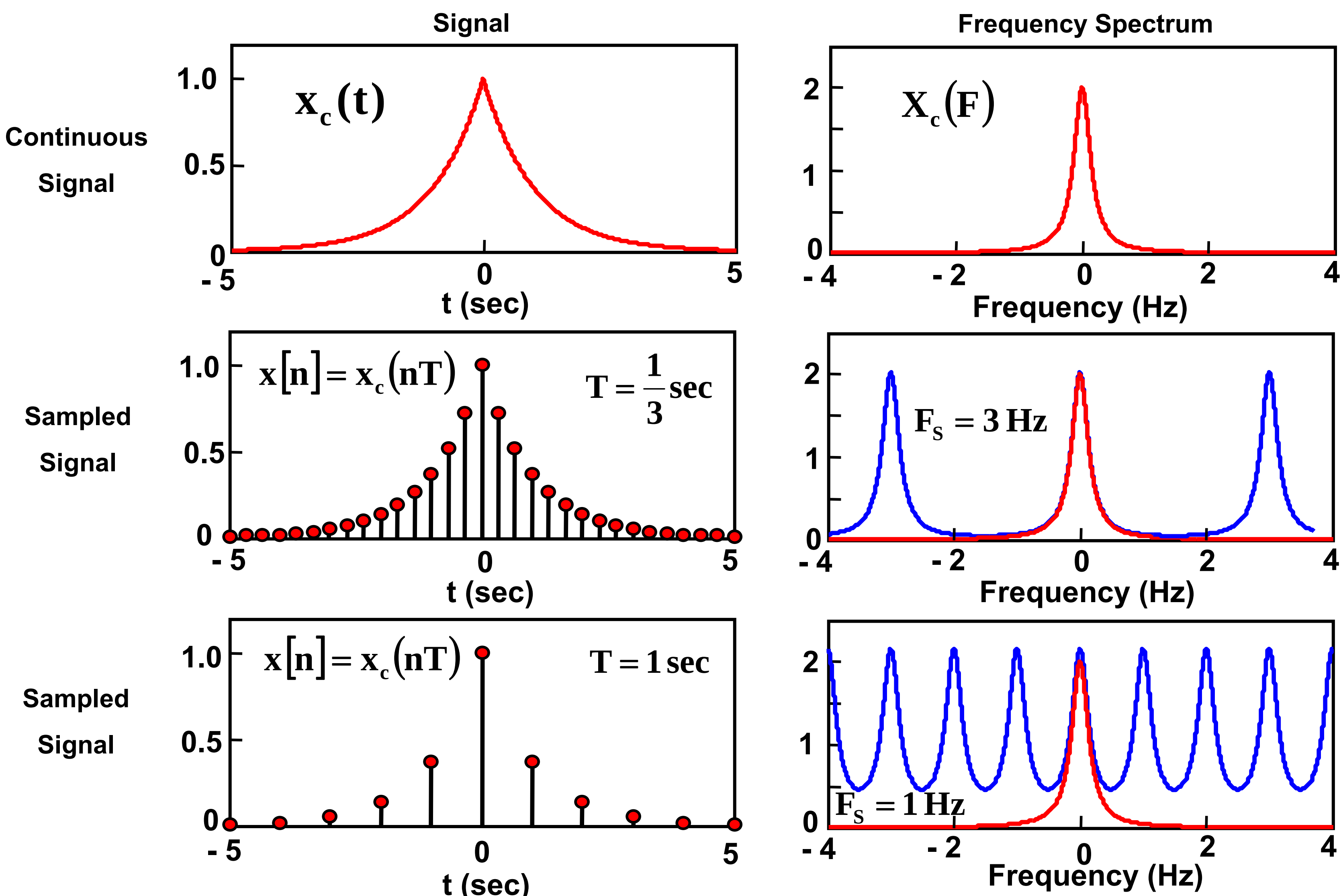
$$X(F) = \frac{1}{T}\sum_{\ell=-\infty}^{\infty} X_c(F - \ell F_c) = \hat{X}_c(F) = \begin{cases} T\,X(F) & |F| \le \dfrac{F_S}{2} \\ 0 & |F| > \dfrac{F_S}{2} \end{cases}$$

Inverse Fourier Transform → $\hat{x}_c(t)$

Reconstructed Signal

Adapted from Proakis and Manolakis, Reference 1

Spectrum of Reconstructed Signal



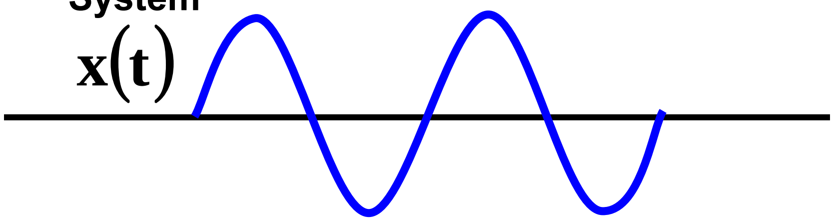
Adapted from Proakis and Manolakis, Reference 1

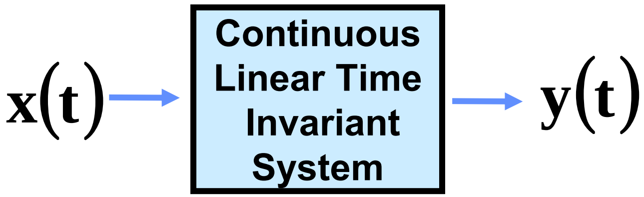
Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
  - General properties
  - A/D Conversion
  - Sampling Theorem and Aliasing
  - Convolution of Discrete Time Signals
  - Fourier Properties of Signals
    - Continuous vs. Discrete
    - Periodic vs. Aperiodic
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

Convolution for Discrete Time Systems

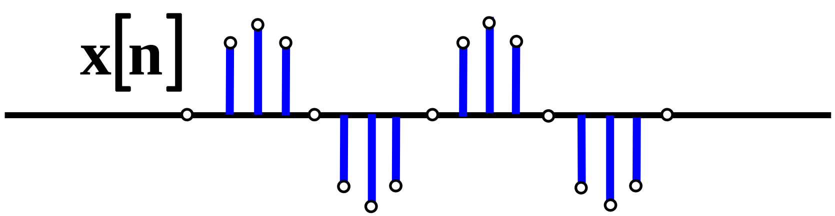
Continuous-time System

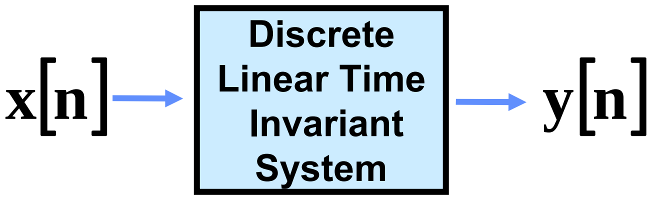




$$y(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau$$

Discrete-time System



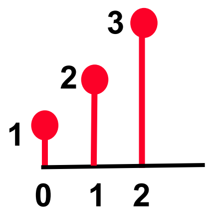


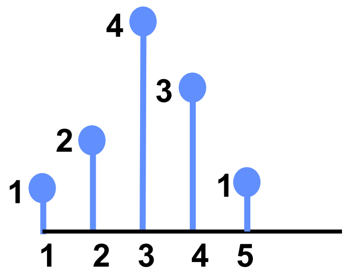
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k]$$

Graphical Implementation of Convolution

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

Example:

$h[k] =$ 

$x[k] =$ 

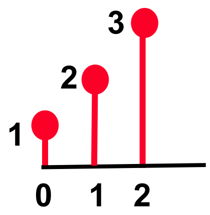
- **Step 1 : Plot the sequences,** $x[k]$ **and** $h[k]$

Graphical Implementation of Convolution

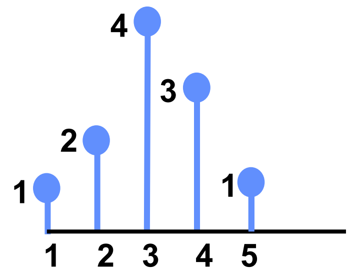
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Example:

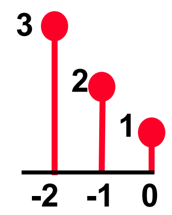
$h[k] =$



$x[k] =$



$h[-k] =$



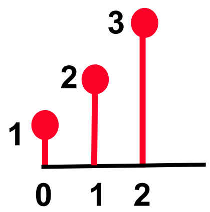
- **Step 2 : Take one of the sequences and time reverse it**

Graphical Implementation of Convolution

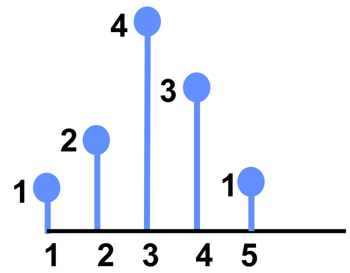
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Example:

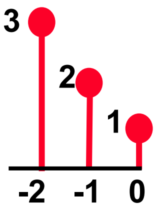
$h[k] =$



$x[k] =$

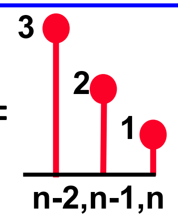


$h[-k] =$



- **Step 3 : Shift** $h[-k]$ **by** n **, yielding**
  - $n < 0$ **a shift to the left**
  - $n > 0$ **a shift to the right**

$h[n-k] =$

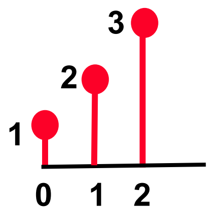


Graphical Implementation of Convolution

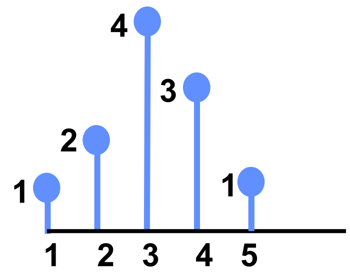
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Example:

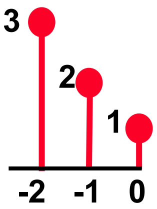
$h[k] =$



$x[k] =$



$h[-k] =$



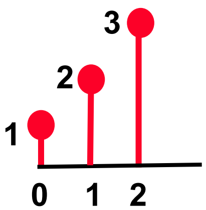
- Step 4 : For each value of n,multiply the sequences $x[k]$ and $h[n-k]$; and add products together for all values of k to produce $y[n]$

Graphical Implementation of Convolution

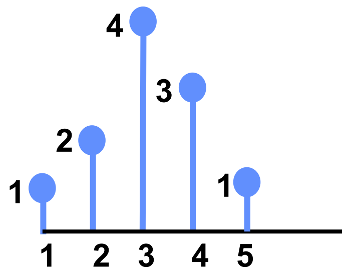
$$y[n]=\sum_{k=-\infty}^{\infty} h[k]x[n-k]=\sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Example:

$h[k]=$

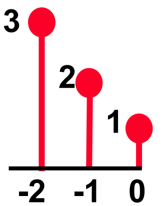


$x[k]=$

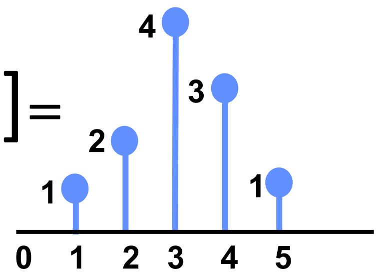


for $n=0$

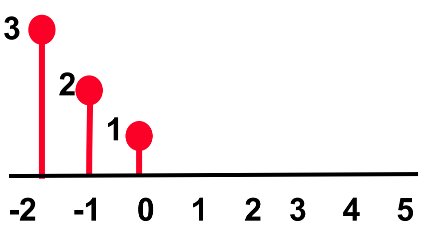
$h[-k]=$

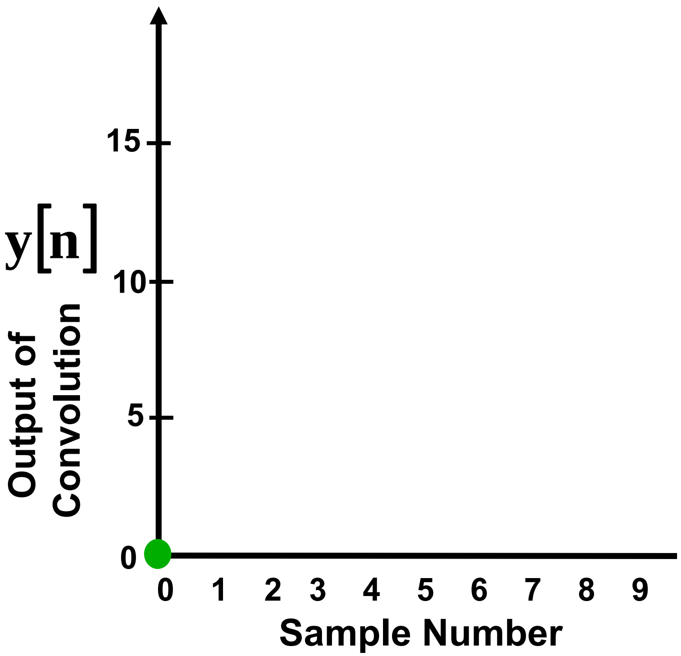


$x[k]=$



$h[-k]=$





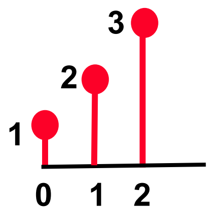
No overlap – $y[n]=0$

Graphical Implementation of Convolution

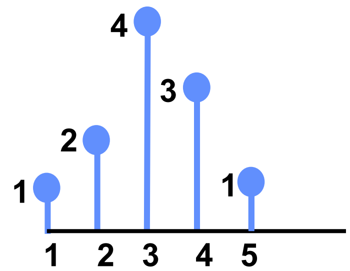
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Example:

$h[k] =$

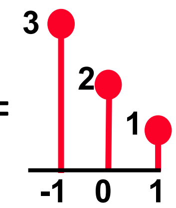


$x[k] =$

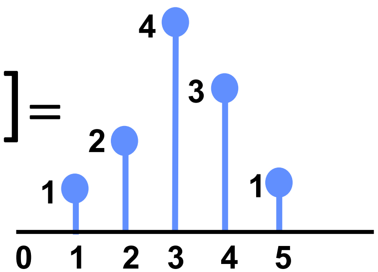


for $n = 1$

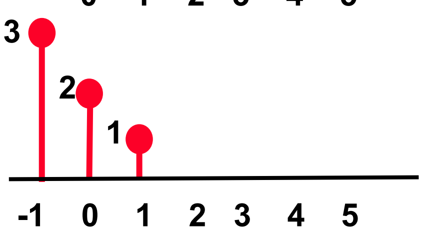
$h[1-k] =$

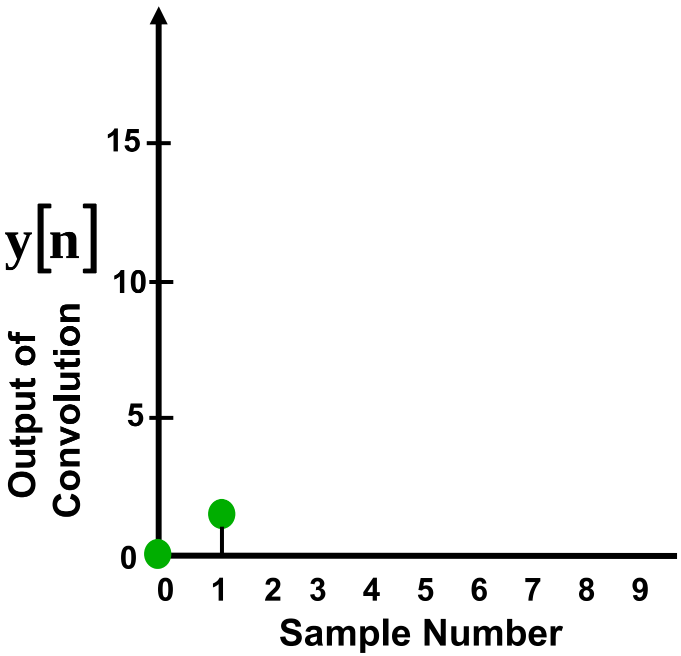


$x[k] =$



$h[1-k] =$



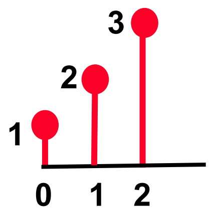


One sample overlaps – $y[n]$ = (1x1) = 1

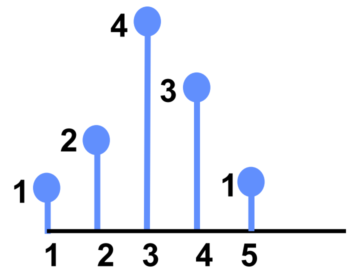
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Example:

$h[k] =$

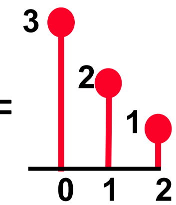


$x[k] =$

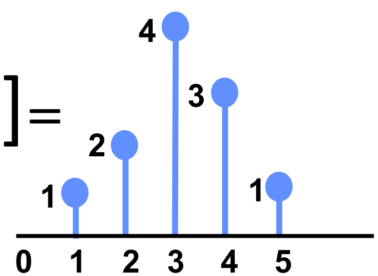


for $n = 2$

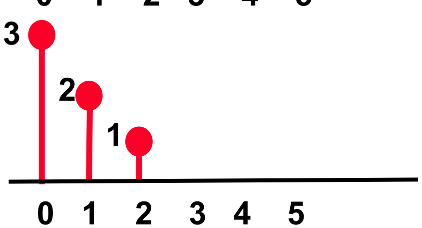
$h[2-k] =$

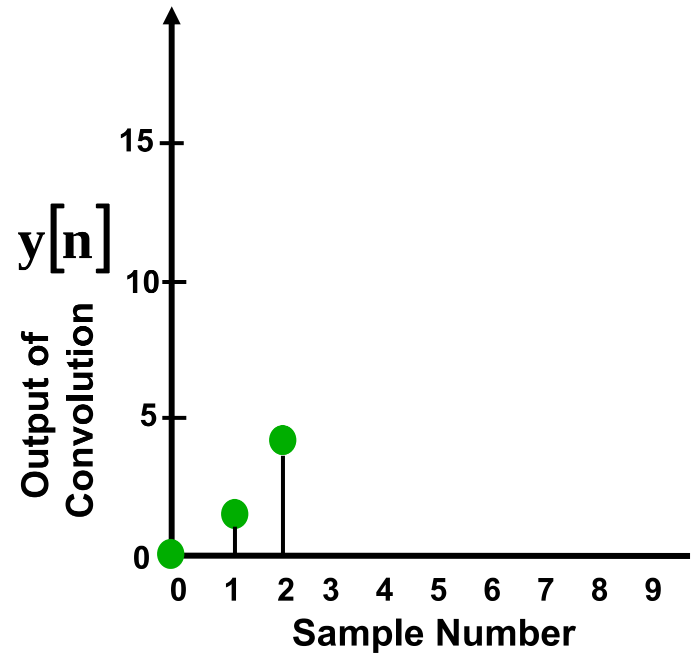


$x[k] =$



$h[2-k] =$





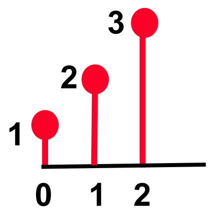
Two samples overlaps – $y[n]$ = (1x2)+(2x1) = 4

Graphical Implementation of Convolution

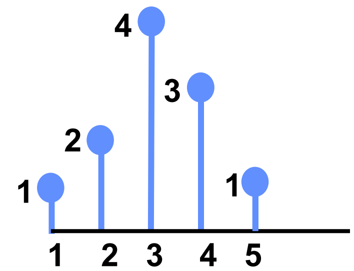
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\,x[n-k] = \sum_{k=-\infty}^{\infty} x[k]\,h[n-k]$$

Example:

$h[k] =$

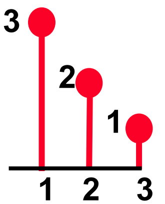


$x[k] =$

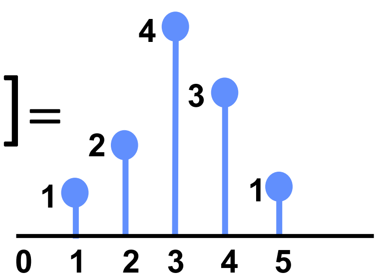


for $n = 3$

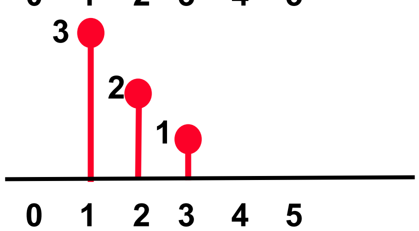
$h[3-k] =$

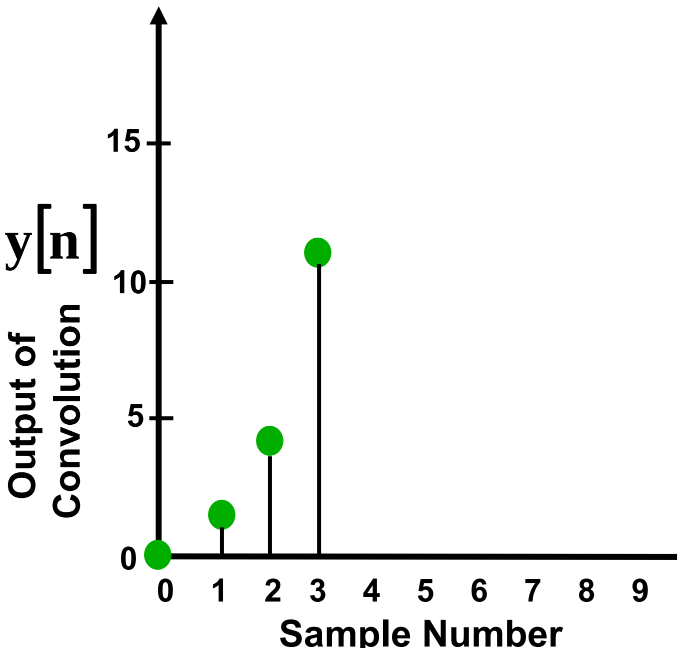


$x[k] =$



$h[3-k] =$





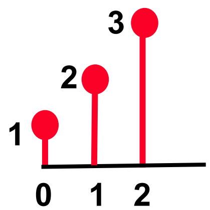
Three samples overlaps – $y[n]$ = (1x3)+(2x2)+(4x1) = 11

Graphical Implementation of Convolution

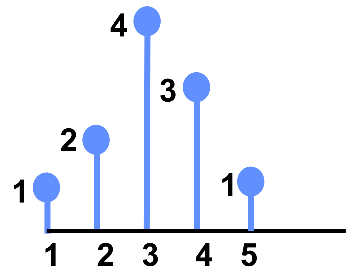
$$y[n]=\sum_{k=-\infty}^{\infty} h[k]x[n-k]=\sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Example:

$h[k]=$

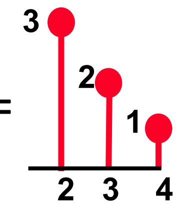


$x[k]=$

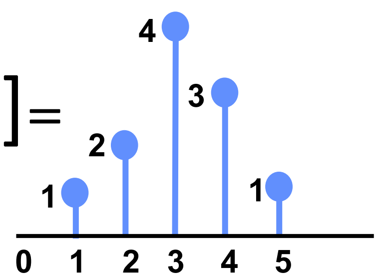


for $n=4$

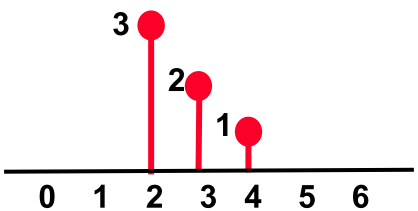
$h[4-k]=$

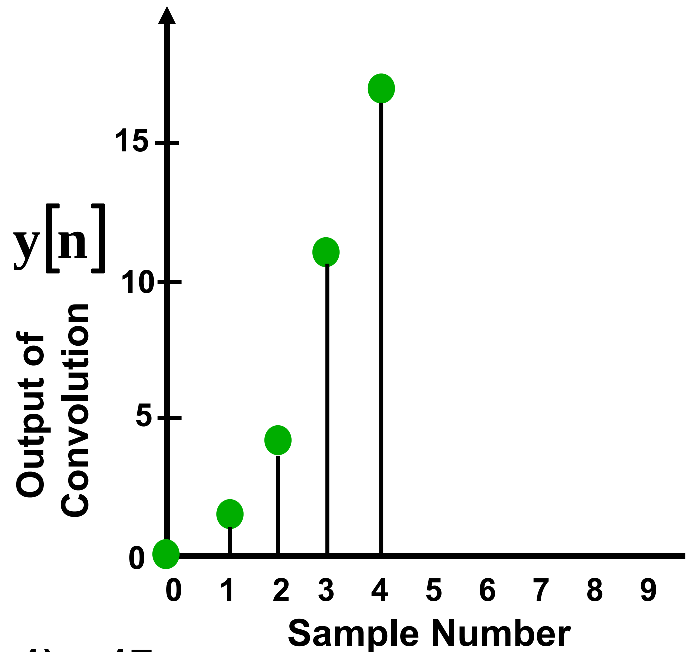


$x[k]=$



$h[4-k]=$





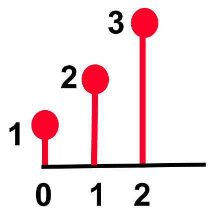
Three samples overlaps – $y[n]$ = (2x3)+(4x2)+(3x1) = 17

Graphical Implementation of Convolution

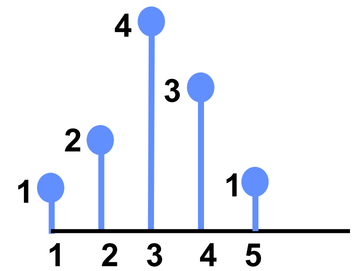
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\,x[n-k] = \sum_{k=-\infty}^{\infty} x[k]\,h[n-k]$$

Example:

$h[k] =$

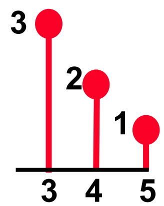


$x[k] =$

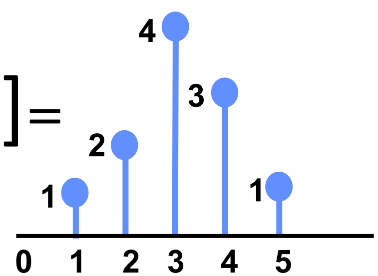


for $n = 5$

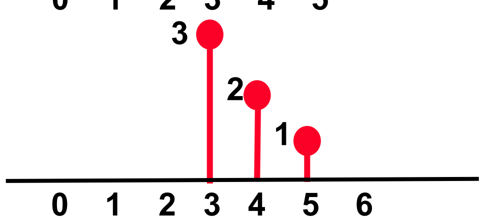
$h[5-k] =$

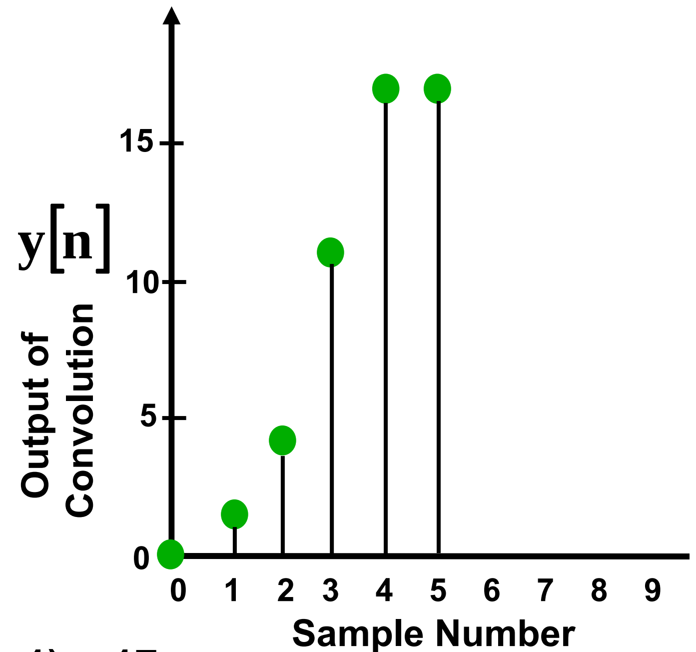


$x[k] =$



$h[5-k] =$





Three samples overlaps – $y[n]$ = (4x3)+(3x2)+(1x1) = 17

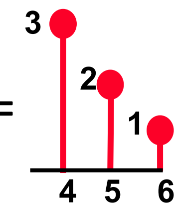
Graphical Implementation of Convolution

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

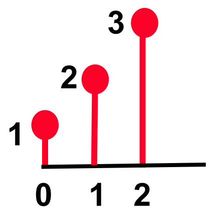
Example:

for $n = 6$

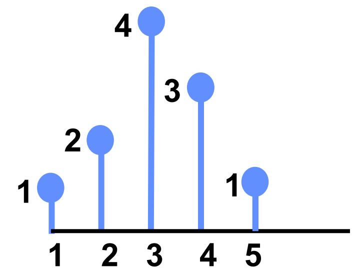
$h[6-k] =$



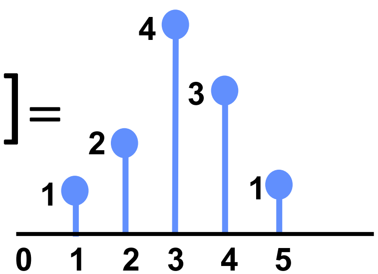
$h[k] =$



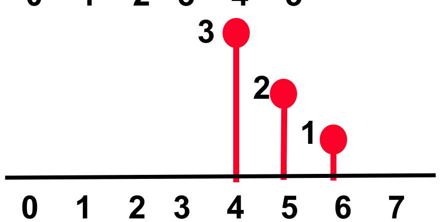
$x[k] =$

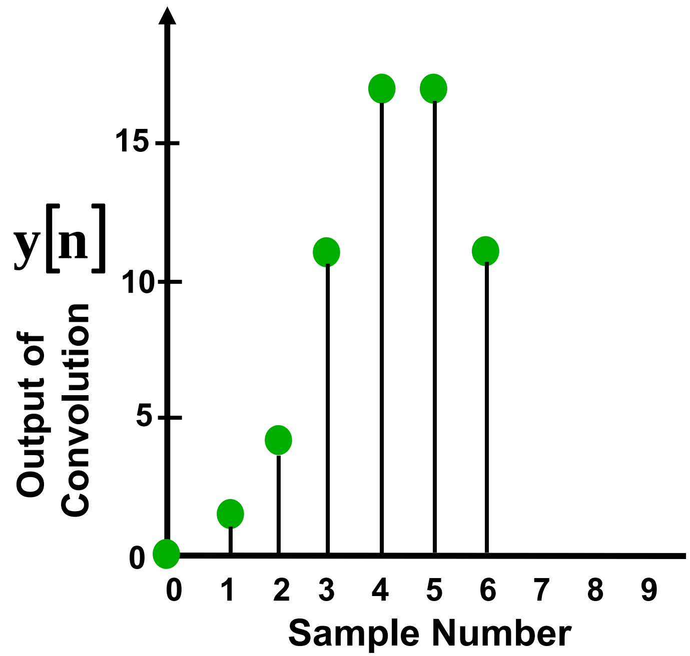


$x[k] =$



$h[6-k] =$





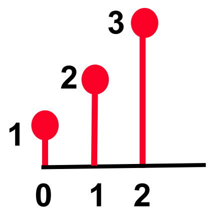
Two samples overlaps – $y[n]$ = (3x3)+(1x2) = 11

Graphical Implementation of Convolution

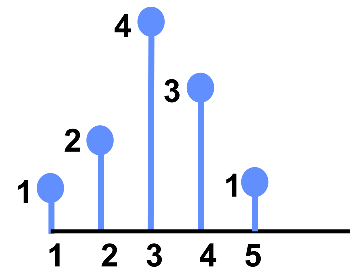
$$y[n]=\sum_{k=-\infty}^{\infty} h[k]x[n-k]=\sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Example:

$h[k]=$

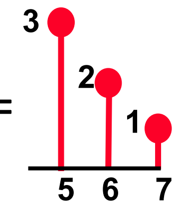


$x[k]=$

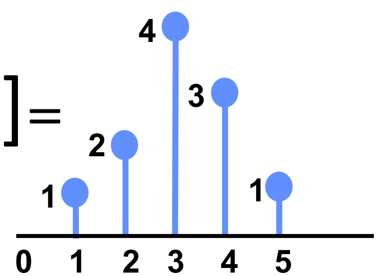


for $n=7$

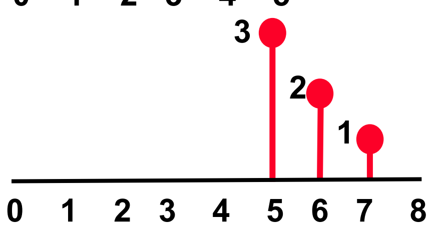
$h[7-k]=$

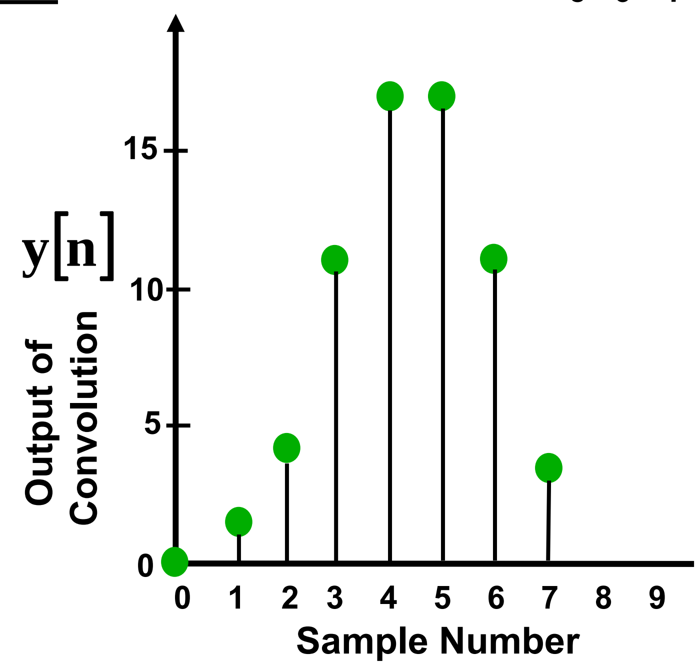


$x[k]=$



$h[7-k]=$





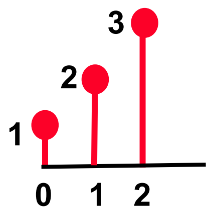
One sample overlaps – $y[n]$ = (1x3) = 3

Graphical Implementation of Convolution

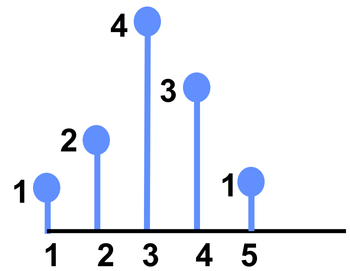
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Example:

$h[k] =$

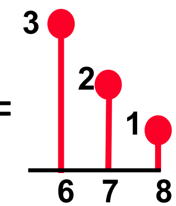


$x[k] =$

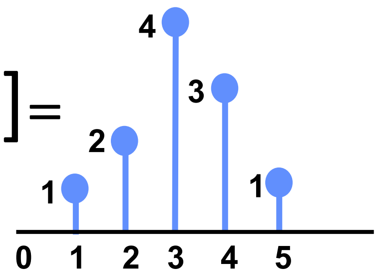


$\text{for} \quad n = 8$

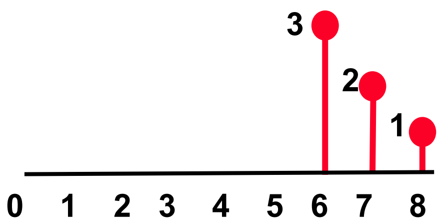
$h[8-k] =$

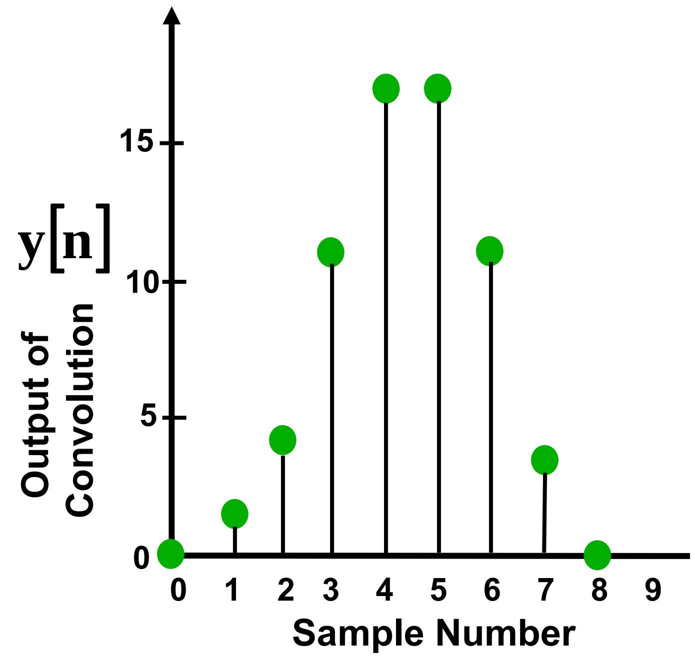


$x[k] =$



$h[8-k] =$





No overlap – $y[n] = 0$

Summary- Linear Discrete Time Systems

- Any Linear and Time-Invariant (LTI) system can be completely described by its impulse response sequence

$$\delta[n] \xrightarrow{H} h[n]$$

- The output of any LTI can be determined using the convolution summation

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

- The impulse response provides the basis for the analysis of an LTI system in the time-domain

- The frequency response function provides the basis for the analysis of an LTI system in the frequency-domain

Adapted from MIT LL Lecture Series by D. Manolakis

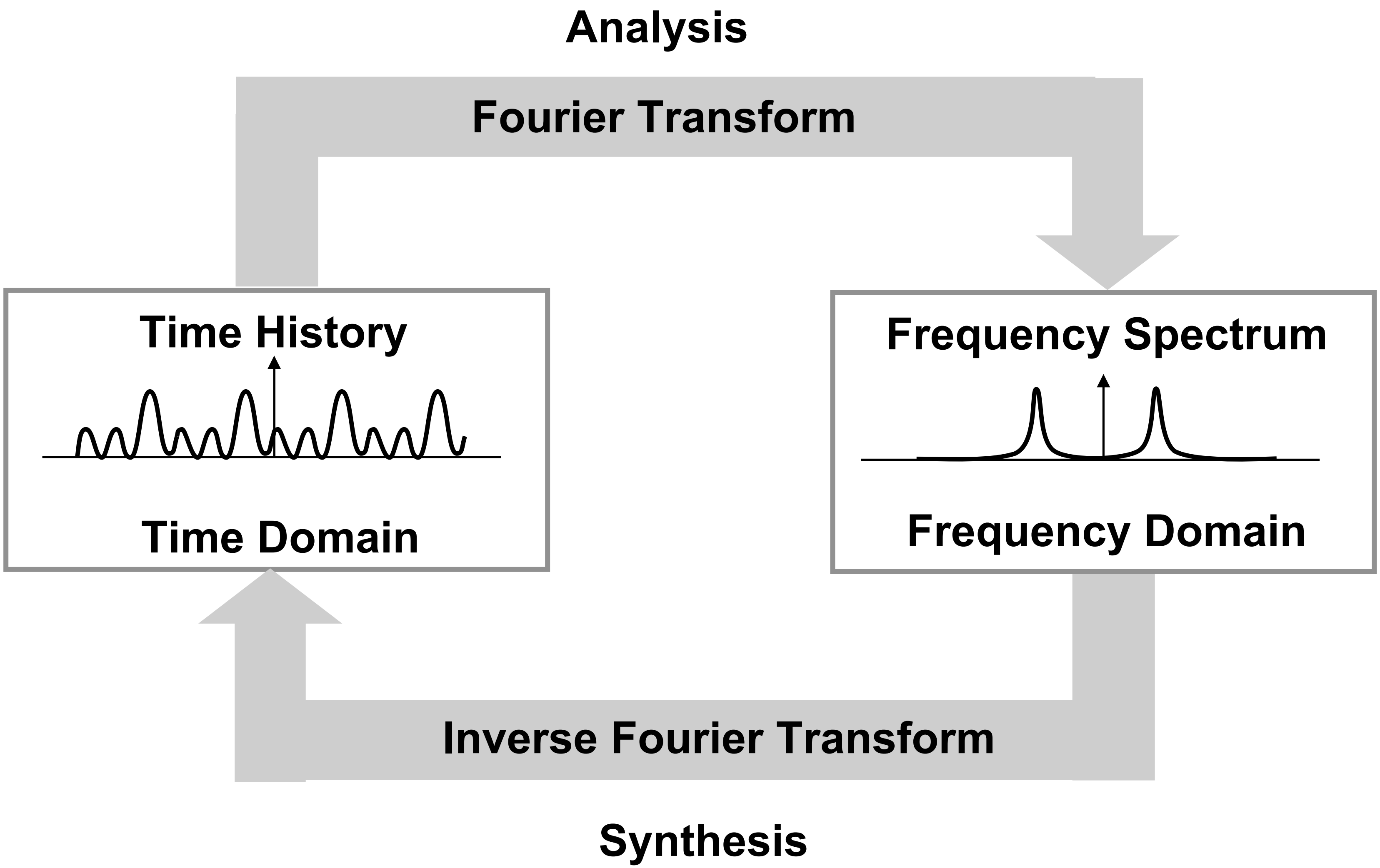
Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
  - General properties
  - A/D Conversion
  - Sampling Theorem and Aliasing
  - Convolution of Discrete Time Signals
  - Fourier Properties of Signals
    - Continuous vs. Discrete
    - Periodic vs. Aperiodic
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

Frequency Analysis of Signals

- Decomposition of signals into their frequency components
  - A series of sinusoids of complex exponentials
- The general nature of signals
  - Continuous or discrete
  - Aperiodic or periodic
- Radar echoes, from each transmitted pulse, are continuous and aperiodic, and are usually transformed into discrete signals by an A/D converter before further processing
  - Complex signals

Time and Frequency Domains

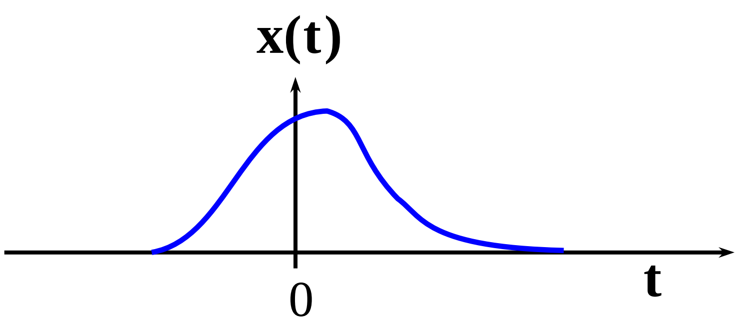


Fourier Properties of Signals

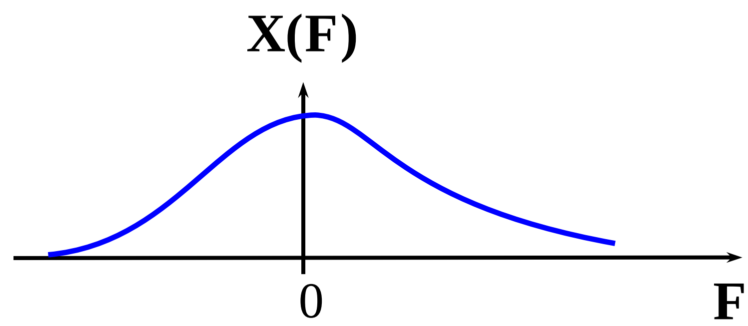
- Continuous-Time Signals
  - Periodic Signals: Fourier Series
  - Aperiodic Signals: Fourier Transform
- Discrete-Time Signals
  - Periodic Signals: Fourier Series
  - Aperiodic Signals: Fourier Transform

Fourier Transform for Continuous-Time Aperiodic Signals

Time Domain
Continuous and Aperiodic Signals



Frequency Domain
Continuous and Aperiodic Signals



$$X(F) = \int_{-\infty}^{\infty} x(t)\, e^{-j 2 \pi F t} dt$$

$$x(t) = \int_{-\infty}^{\infty} X(F)\, e^{j 2 \pi F t} dF$$

Adapted from Manolakis et al, Reference 1

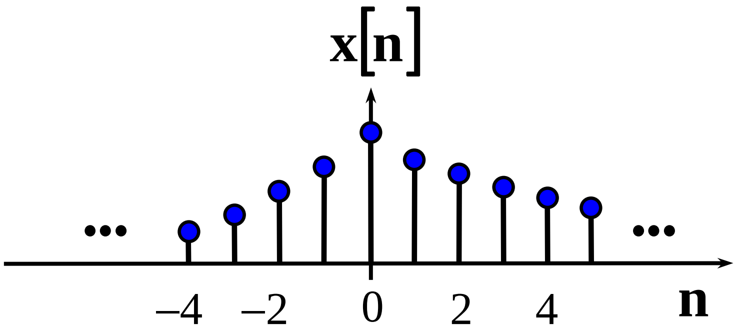
Fourier Properties of Signals

- **Continuous-Time Signals**
  - Periodic Signals: Fourier Series
  - Aperiodic Signals: Fourier Transform

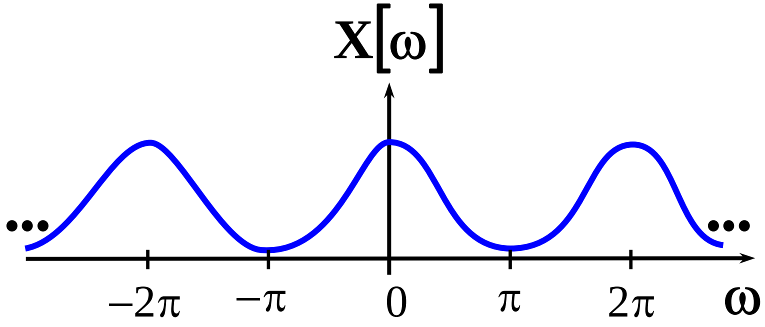
- **Discrete-Time Signals**
  - Periodic Signals: Fourier Series
  - Aperiodic Signals: Fourier Transform

Fourier Transform for Discrete-Time Aperiodic Signals

Time Domain
Discrete and Aperiodic Signals



Frequency Domain
Continuous and Periodic Signals



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

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

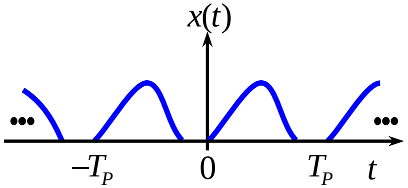
Adapted from Malolakis et al, Reference 1

Summary of Time to Frequency Domain Properties

Continuous- Time Signals

Periodic Signals – Fourier Series

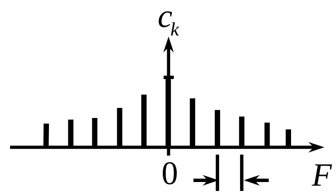
Time-Domain



$$c_k = \frac{1}{T_P}\int_{T_P} x(t) e^{-j2\pi k F_0 t} dt$$

Continuous and Periodic

Frequency-Domain



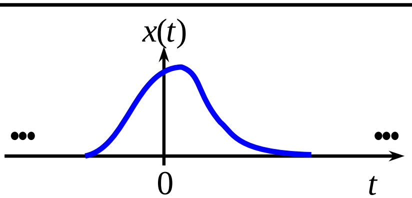
$$F_0 = \frac{1}{T_P}$$

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t}$$

Discrete and Aperiodic

Aperiodic Signals – Fourier Transforms

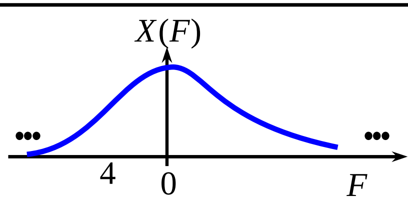
Time-Domain



$$X(F) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi F t} dt$$

Continuous and Aperiodic

Frequency-Domain



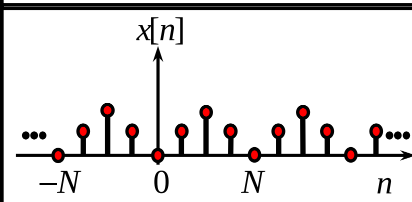
$$x(t) = \int_{-\infty}^{\infty} X(F) e^{j2\pi F t} dF$$

Continous and Aperiodic

Discrete- Time Signals

Periodic Signals – Fourier Series

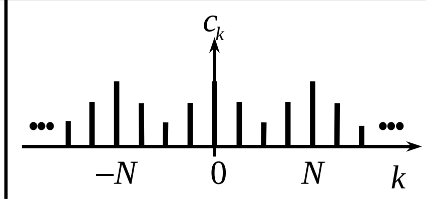
Time-Domain



$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Discrete and Periodic

Frequency-Domain

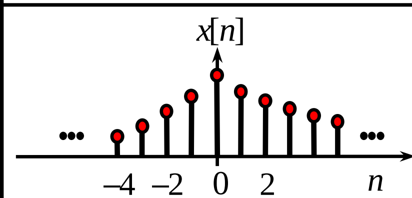


$$x[n] = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{N}kn}$$

Discrete and Periodic

Aperiodic Signals – Fourier Transforms

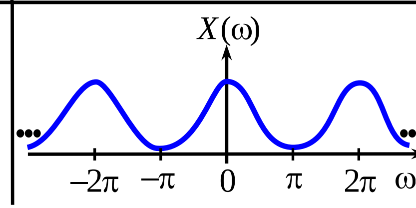
Time-Domain



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

Discrete and Aperiodic

Frequency-Domain



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

Continous and Periodic

Adapted from Proakis and Manolakis, Reference 1

Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
- Discrete Fourier Transform (DFT)
  - Calculation
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

Direct DFT Computation

Aka "Twiddle Factor"

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{kn} \qquad 0 \le k \le N-1$$

$$W_N^{kn} = e^{-2\pi j k n / N}$$

$$X_R[k] = \sum_{n=0}^{N-1} \left\{ x_R[n]\cos\left(\frac{2\pi}{N} k n\right) + x_I[n]\sin\left(\frac{2\pi}{N} k n\right) \right\}$$

$$X_I[k] = -\sum_{n=0}^{N-1} \left\{ x_R[n]\sin\left(\frac{2\pi}{N} k n\right) - x_I[n]\cos\left(\frac{2\pi}{N} k n\right) \right\}$$

- 1. $2N^2$ evaluations of trigonometric functions
- 2. $4N^2$ real (N^2 complex) multiplications
- 3. $4N(N-2)$ real ($N(N-1)$ complex) additions
- 4. A number of indexing and addressing operations

$\approx N^2$ Complex MADS

MADS
Multiply
And
Divides

Adapted from MIT LL Lecture Series by D. Manolakis

Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

Fast Fourier Transform (FFT)

- An algorithm for each efficiently computing the Discrete Fourier Transform (DFT) and its inverse
- DFT $O\left(N^2\right)$ MADS (Multiplies and Divides)
- FFT $O\left(\frac{N}{2}\log_2 N\right)$ MADS
- FFT algorithm Development - Cooley / Tukey (1965) Gauss (1805)
- Many variations and efficiencies of the FFT algorithm exist
  - Decimation in Time (input - bit reversed, output - natural order)
  - Decimation in Frequency (input - natural order, output - bit reversed)
- The FFT calculation is broken down into a number of sequential stages, each stage consisting of a number of relatively small calculations called "Butterflies"

Radix 2 Decimation in Time FFT Algorithm

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-2\pi j k n/N} = \sum_{n=0}^{N-1} x[n]\, W_N^{kn} \quad 0 \le k \le N-1 \quad W_N^{kn} = e^{-2\pi j k n/N}$$

- Divide DFT of size N into two interleaved DFTs, each of size N/2
  - Example will be $N = 2^3 = 8$
  - Input to each DFT are even and odd $x[n]$s , respectively
- Solve each stage recursively, until the size of the stage's DFT is 2.

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} = \sum_{n\ \text{Even}} x[n] W_N^{nk} + \sum_{n\ \text{Odd}} x[n] W_N^{nk}$$

$$= \sum_{l=0}^{\frac{N}{2}-1} g[l] W_N^{lk} + \sum_{l=0}^{\frac{N}{2}-1} h[l] W_N^{(2l+1)k} = \sum_{l=0}^{\frac{N}{2}-1} g[l] W_{N/2}^{lk} + W_N^{k} \sum_{l=0}^{\frac{N}{2}-1} h[l] W_{N/2}^{lk}$$

N/2 point DFT of $g[l] = G[k]$

Even index and odd index terms of $x[n]$

N/2 point DFT of $h[l] = H[k]$

Radix 2 Decimation in Time FFT Algorithm (continued)

$$X[k] = G[k] + W_N^{nk}\, H[k]$$

- **Using the periodicity of the complex exponentials:**

$$G[k] = G\left[k + \frac{N}{2}\right] \qquad\qquad H[k] = H\left[k + \frac{N}{2}\right]$$

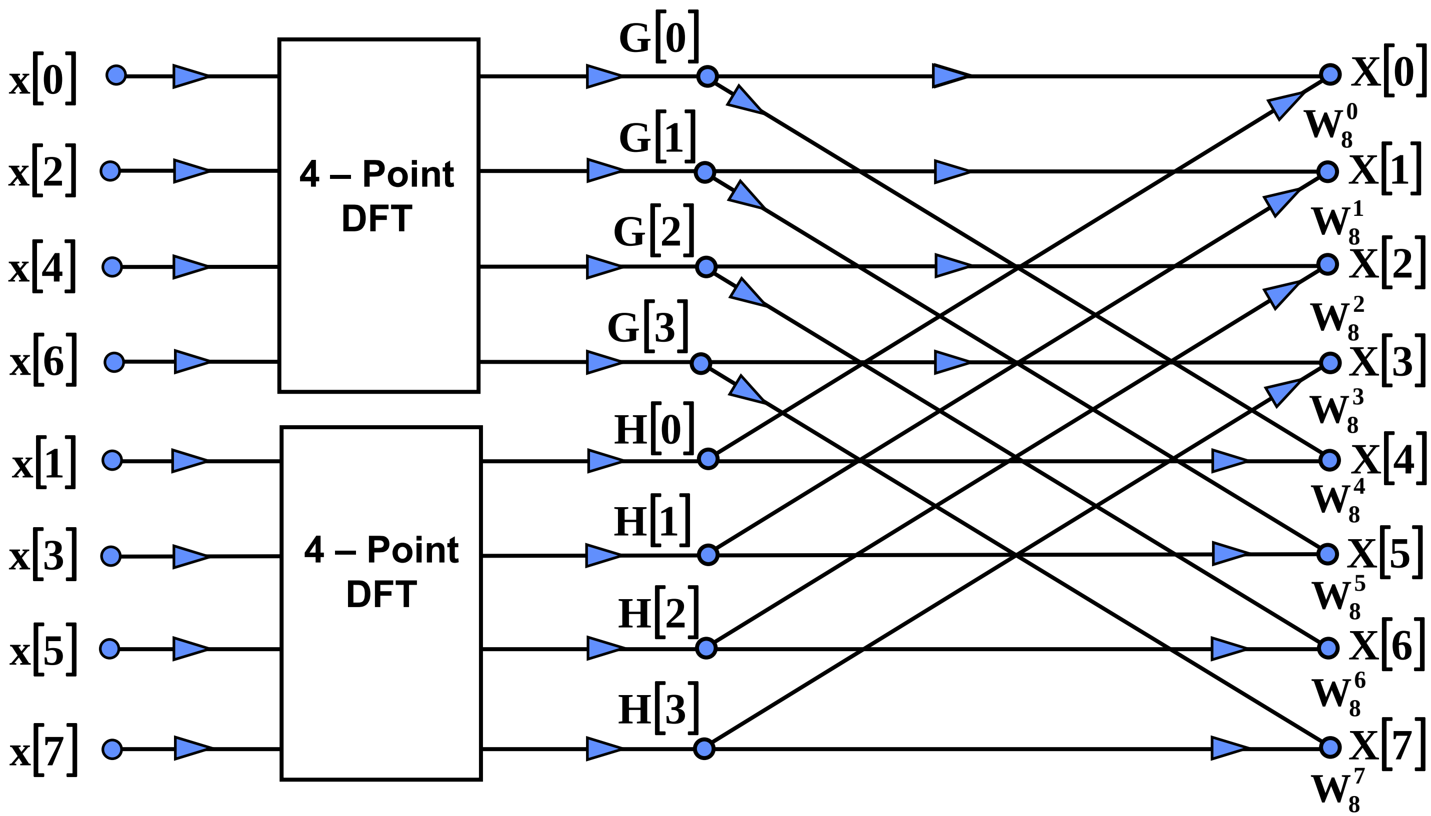
- **And the following properties of the "twiddle factors":**

$$W_N^{k+(N/2)} = W_N^{k}\, W_N^{N/2} = -W_N^{k}$$

then $$W_N^{k+(N/2)}\, H\big(k + (N/2)\big) = -W_N^{k}\, H(k)$$

- **A block diagram of this computational flow is graphically illustrated in the next chart for an 8 point FFT**

8 Point Decimation in Time FFT Algorithm (After First Decimation)



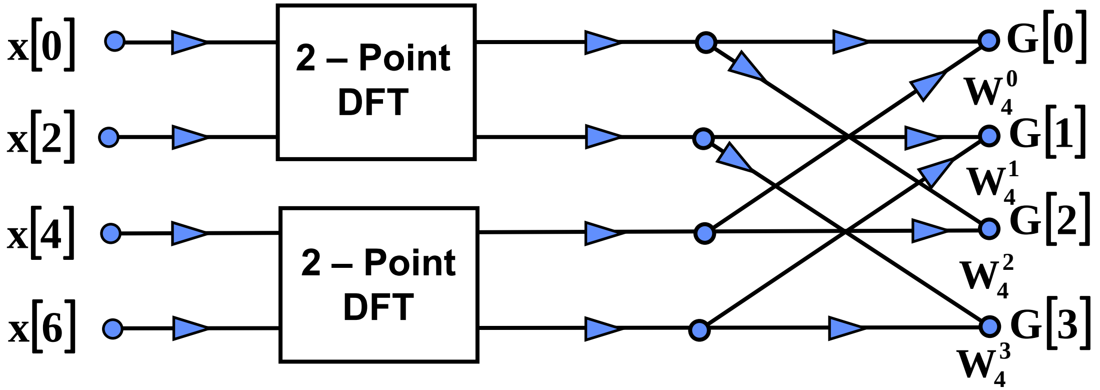
Decimation of 4 Point into two 2 point DFTs

- If N/2 is even, $g[n]$ and $h[n]$ may again be decimated

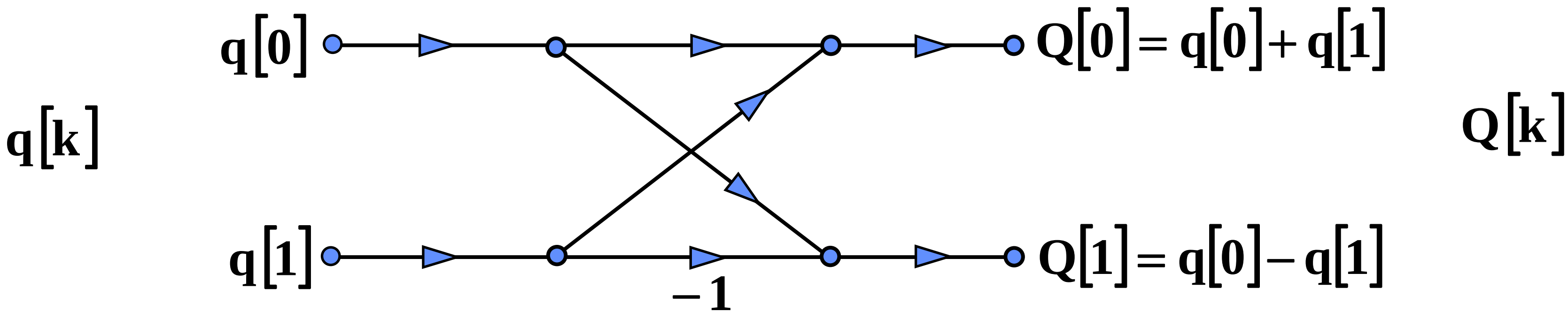
$$G[k] = \sum_{n=0}^{\frac{N}{2}-1} g[n] W_{N/2}^{nk} = \sum_{n\ \text{Even}}^{\frac{N}{2}-1} g[n] W_{N/2}^{nk} + \sum_{n\ \text{Odd}}^{\frac{N}{2}-1} g[n] W_{N/2}^{nk}$$

- This leads to:

$$G[k] = \sum_{n=0}^{\frac{N}{4}-1} g[2n] W_{N/4}^{nk} + W_{N/2}^{k} \sum_{n=0}^{\frac{N}{2}-1} g[2n+1] W_{N/4}^{nk}$$

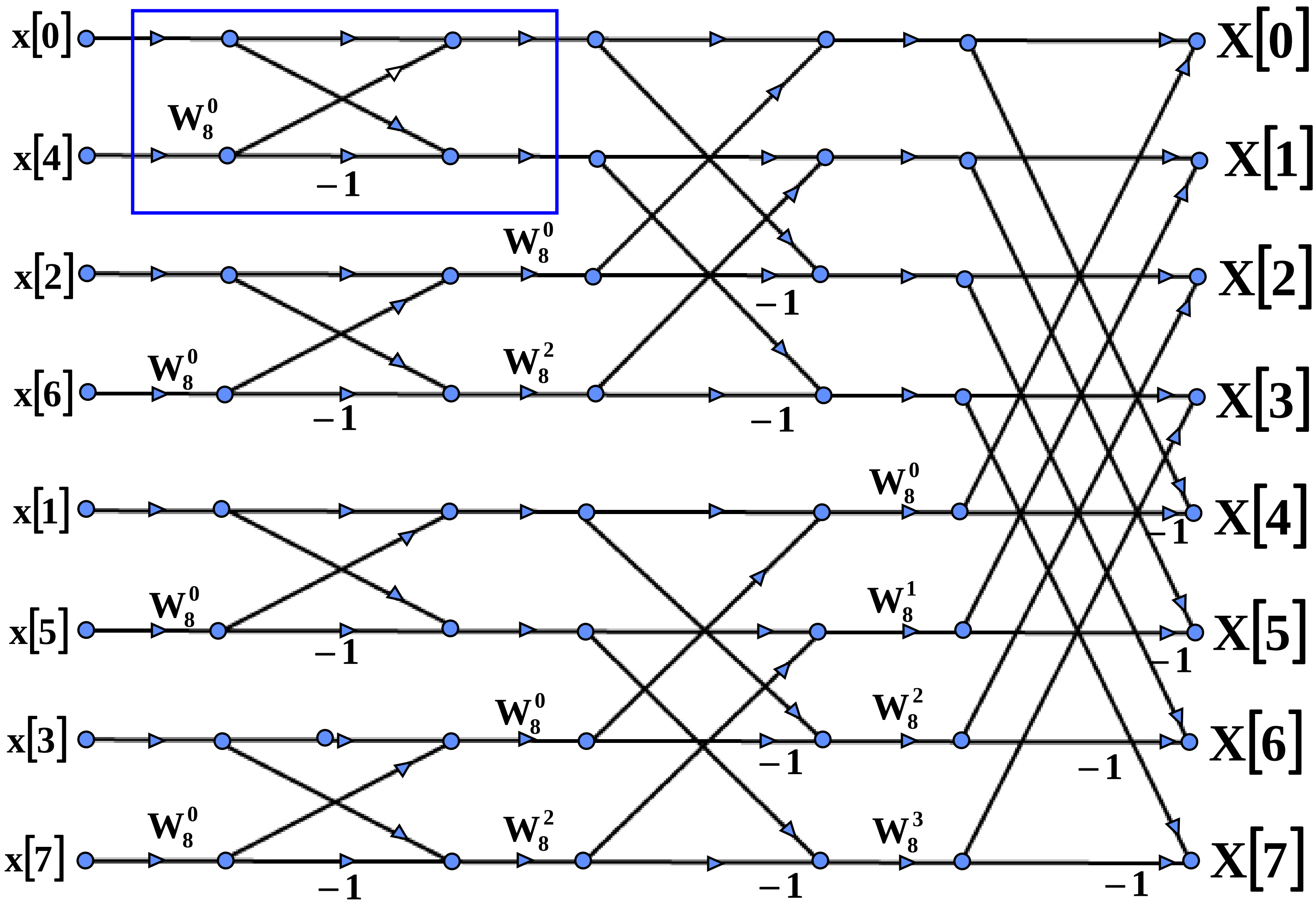


Butterfly for 2 Point DFT



Now, Putting it all together…..

Flow of 8-Point FFT
(Radix 2 - Decimation in Time Algorithm)

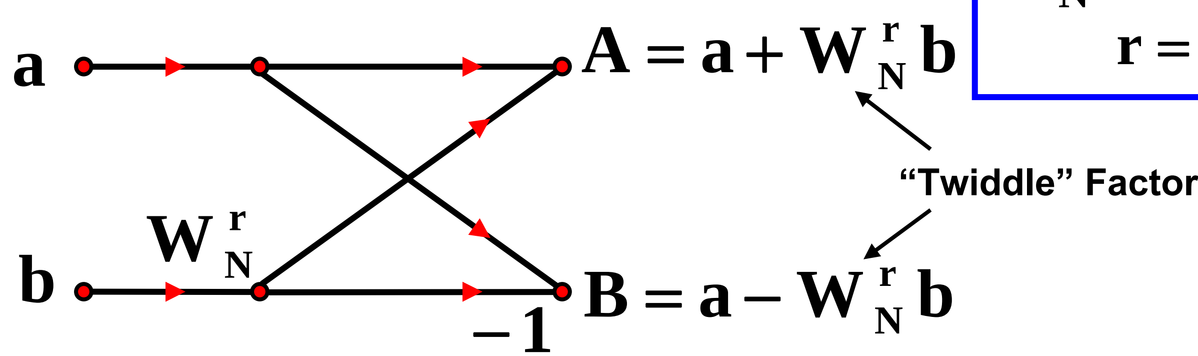


IEEE New Hampshire Section
IEEE AES Society

Basic FFT Computation Flow Graph

$$W_N^{kn} = e^{-2\pi j k n / N}$$
$$r = kn$$

"Butterfly"



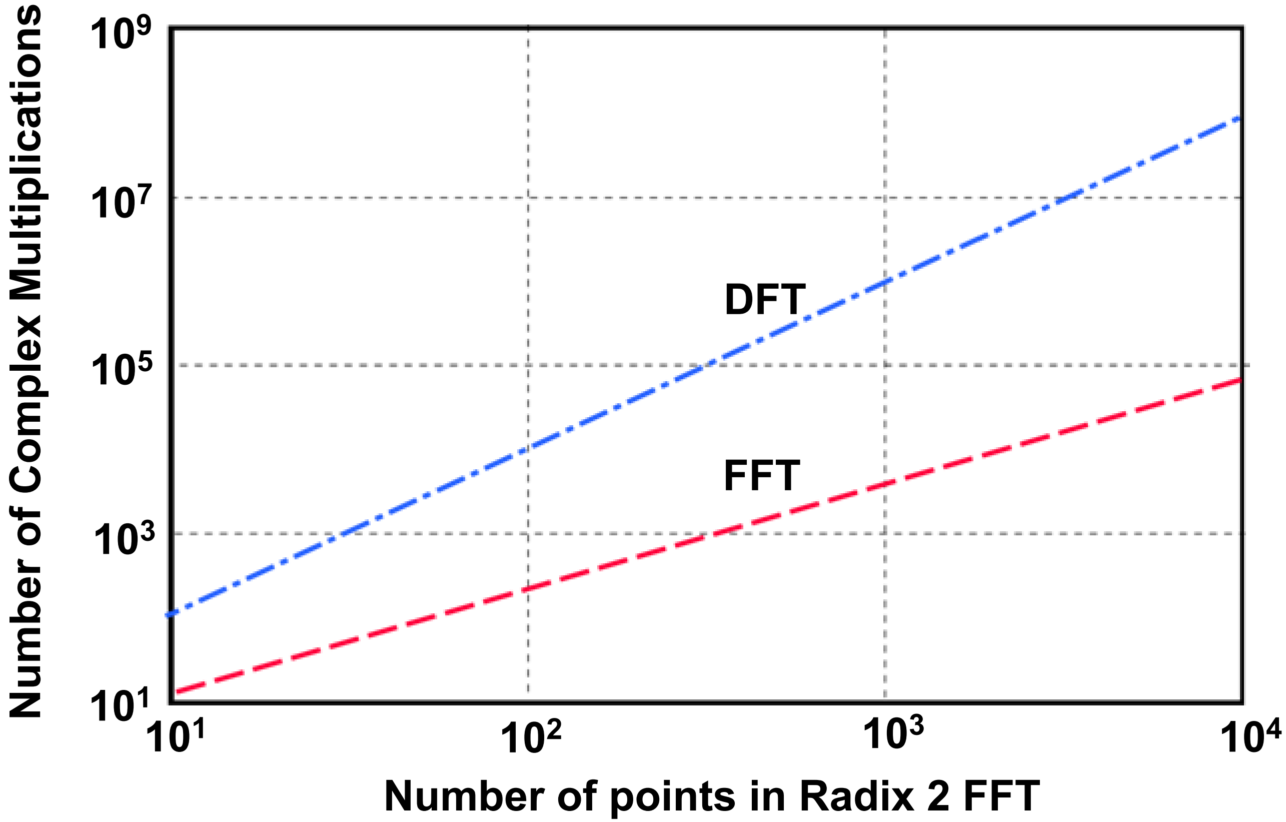
Check over

- Each "Butterfly" takes 2 MADS (Multiplies and Adds)
- Twiddle Factors (For 8 point FFT)

$$W_8^0 = e^{-0} = 1 \qquad W_8^1 = e^{-2\pi j/8} = e^{-\pi j/4} = (1 - j)/\sqrt{2}$$
$$W_8^2 = e^{-\pi j/2} = -j \qquad W_8^3 = e^{-3\pi j/4} = (-1 - j)/\sqrt{2}$$

- 12 Butterflies implies 12 MADS vs. 64 MADS for 8 point DFT
- 512 point FFT more than 100 times faster than 512 DFT

Computational Speed – DFT vs. FFT



Lines Drawn Through Data Points

- Discrete Fourier Transform (O ~ N^2)
- Fast Fourier Transform (O ~ N $\log_2$ N)

Adapted from Lyons, Reference 2

Fast Fourier Transform (FFT) - Summary

- **Fast Fourier Transform (FFT) algorithms make possible the computation of DFT with** $O\,((N/2)\,\log_2 N)$ **MADS as opposed to** $O\,N^2$ **MADS**

- **Many other implementations of the FFT exist:**
  - **Radix 2 decimation in frequency algorithm**
  - **Radar-Brenner algorithm**
  - **Bluestein's algorithm**
  - **Prime Factor algorithm**

- **The details of FFT algorithms are important to the designers of real-time DSP systems in software or hardware**

- **An interesting history of FFT algorithms**
  - **Heideman, Johnson, and Burrus, "*Gauss and the History of FFT*," IEEE ASSP Magazine, Vol. 1, No. 4, pp. 14-21, October 1984**

Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

Finite and Infinite Response Filters

- Infinite Impulse Response (IIR) Filters
  - Output of filter depends on past time history $\left(-\infty\right)$
  - Example :

$$y\left[n\right]=\frac{1}{M}x\left[n\right]+\frac{M-1}{M}y\left[n-1\right]$$

- Finite Impulse Response (FIR) Filters
  - Output depends on the finite past
  - Example: DFT

$$X\left[k\right]=\sum_{n=0}^{N-1}x\left[n\right]e^{-2\pi j k n/N}$$

  - Other examples:

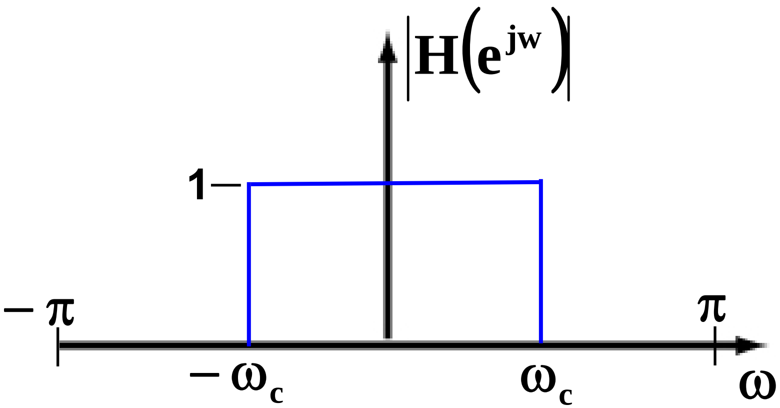
$$y\left[k\right]=\sum_{n=0}^{N-1}a\left[k,n\right]x\left[n\right]x\left[1\right]$$

or

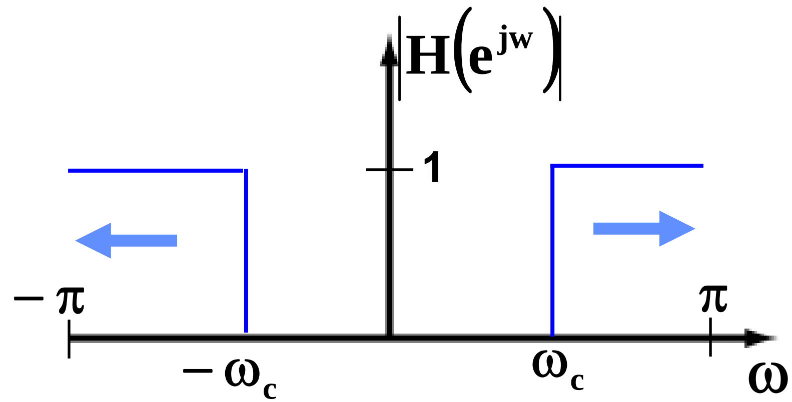
$$y\left[n\right]=x\left[n,2\right]-x\left[n,1\right]$$

Four Basic Filter Types- An Idealization

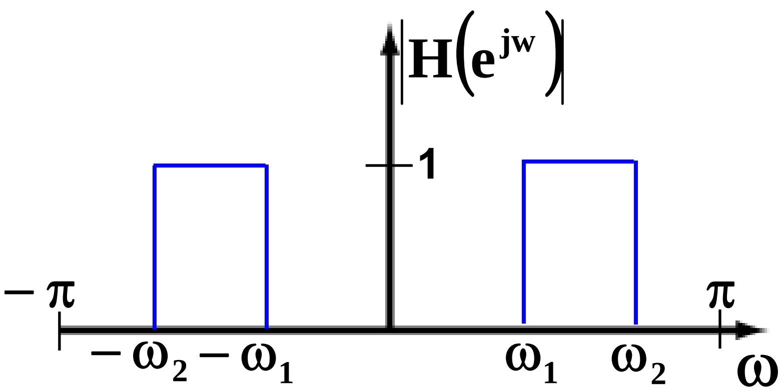
Ideal Low Pass Filter



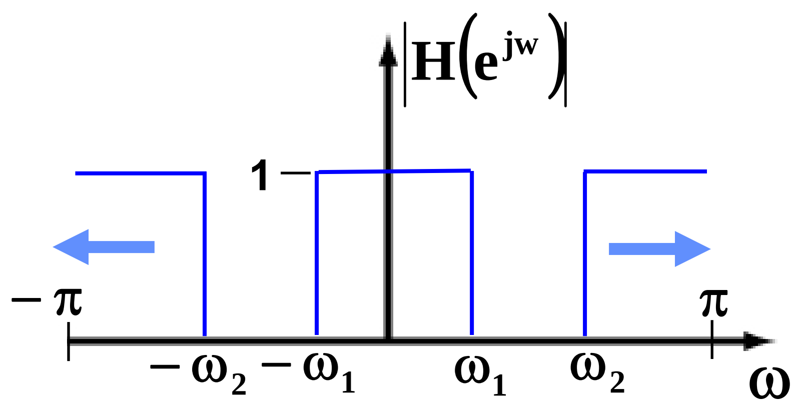
Ideal High Pass Filter



Ideal Bandpass Filter



Ideal Bandstop Filter



Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters

Windowing / Weighting of Filters

- If we take a square pulse, sample it M times, and calculate the Fourier transform of this uniform rectangular "window":

$$W(\omega) = \sum_{n=0}^{M-1} e^{-j\omega n} = \frac{1 - e^{-j\omega M}}{1 - e^{-j\omega}} = e^{-j(M-1)/2} \frac{\sin(\omega M/2)}{\sin(\omega/2)}$$

$$|W(\omega)| = \frac{|\sin(\omega M/2)|}{|\sin(\omega/2)|} \qquad -\pi \leq \omega \leq \pi$$

- This is recognized as the sinc function which has 13 dB sidelobes

- If lower sidelobes are needed , at the cost of a widened pass band, one can multiply the elements of the pulse sequence with one of a number of weighting functions, which will adjust the sidelobes appropriately

Commonly Used Window Functions

- **Rectangular**

$$w[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

- **Bartlett (triangular)**

$$w[n] = \begin{cases} 2n/M, & 0 \le n \le M/2 \\ 2 - 2n/M & M/2 < n \le M \\ 0, & \text{otherwise} \end{cases}$$

- **Hanning**

$$w[n] = \begin{cases} 0.5 - 0.5\cos(2\pi n/M), & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

- **Hamming**

$$w[n] = \begin{cases} 0.54 - 0.46\cos(2\pi n/M), & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

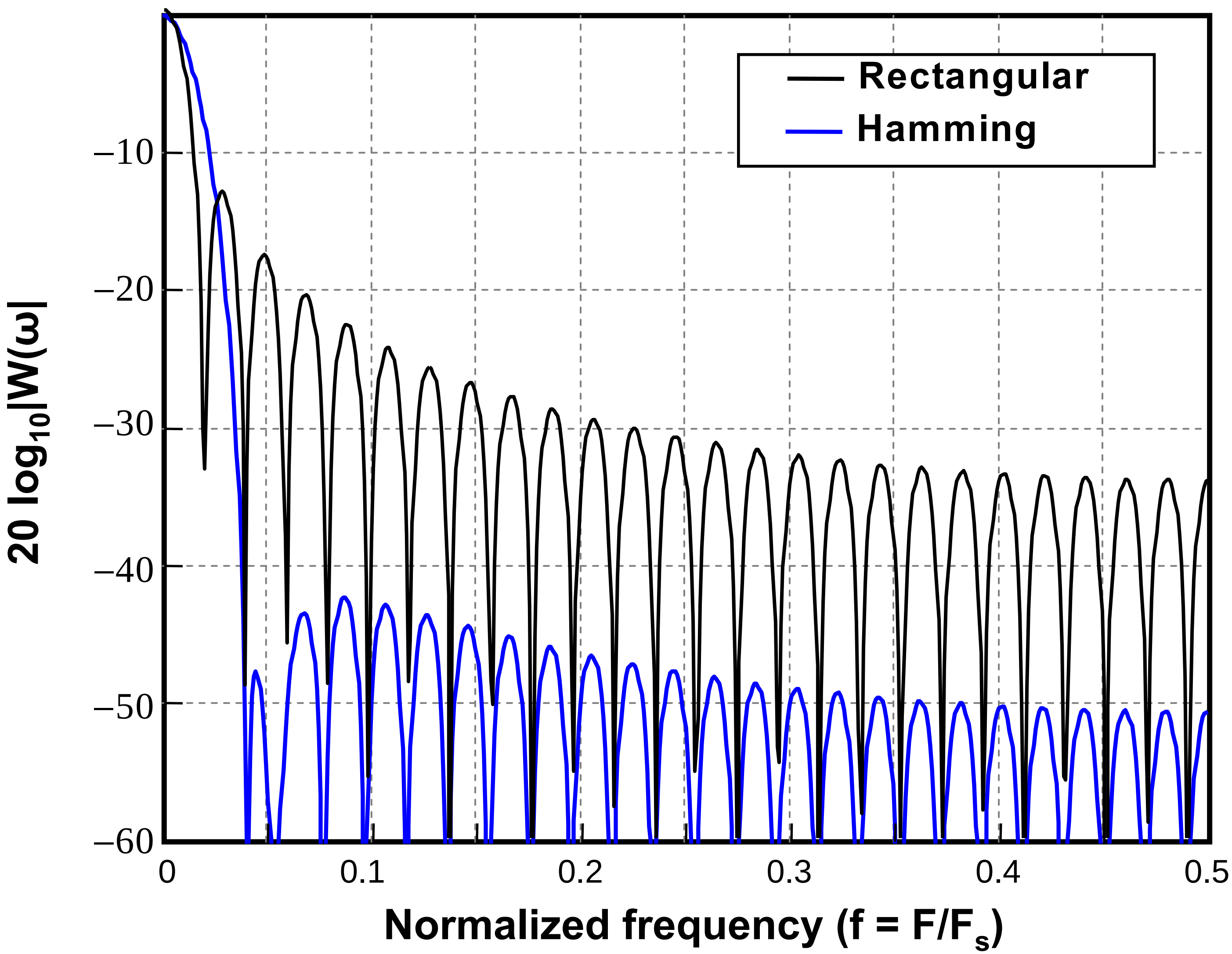
- **Blackman**

$$w[n] = \begin{cases} 0.42 - 0.5\cos(2\pi n/M) + 0.08\cos(4\pi n/M), & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

Comparison of Common Windows

Type of Window	Peak Sidelobe Amplitude (dB)	Approximate Width of Main Lobe
Rectangular	-13	$4\pi/(M+1)$
Bartlett (triangular)	-25	$8\pi/M$
Hanning	-31	$8\pi/M$
Hamming	-41	$8\pi/M$
Blackman	-57	$12\pi/M$

Comparison of Rectangular & Hamming Windows



Summary

- **A brief review of the prerequisite Signal & Systems, and Digital Signal Processing knowledge base for this radar course has been presented**
  - **Viewers requiring a more in depth exposition of this material should consult the references at the end of the lecture**

- **The topics discussed were:**
  - **Continuous signals and systems**
  - **Sampled data and discrete time systems**
  - **Discrete Fourier Transform (DFT)**
  - **Fast Fourier Transform (FFT)**
  - **Finite Impulse Response (FIR) filters**
  - **Weighting of filters**

References

1. Proakis, J, G. and Manolakis, D. G., *Digital Signal Processing, Principles, Algorithms, and Applications,* Prentice Hall, Upper Saddle River, NJ, 4th Ed., 2007
2. Lyons, R. G., *Understanding Digital Signal Processing*, Prentice Hall, Upper Saddle River, NJ, 2nd Ed., 2004
3. Hsu, H. P., *Signals and Systems*, McGraw Hill, New York, 1995
4. Hayes, M. H., *Digital Signal Processing*, McGraw Hill, New York, 1999
5. Oppenheim, A. V. et al, *Discrete Time Signal Processing*, Prentice Hall, Upper Saddle River, NJ, 2nd Ed., 1999
6. Boulet, B., *Fundamentals of Signals and Systems*, Prentice Hall, Upper Saddle River, NJ, 2nd Ed., 2000
7. Richards, M. A., *Fundamentals of Radar Signal Processing,* McGraw Hill, New York, 2005
8. Skolnik, M., *Radar Handbook*, McGraw Hill, New York, 2nd Ed., 1990
9. Skolnik, M., *Introduction to Radar Systems*, McGraw Hill, New York, 3rd Ed., 2001

Acknowledgements

- Dr Dimitris Manolakis
- Dr. Stephen C. Pohlig
- Dr William S. Song

Homework Problems

- From Proakis and Manolakis, Reference 1
  - Problems 2.1, 2.17, 4.9a and b, 4.10 a and b, 6.1, 6.9 a and b, 8.1 and 8.8
- Or
- And from Hays, Reference 4
  - Problems 1.41, 1.49, 1.54, 1.59, 2.46, 2.57, 2.58, 3.27, 3.28, 3.34, 6.44, 6.45