Digital Signal Processing

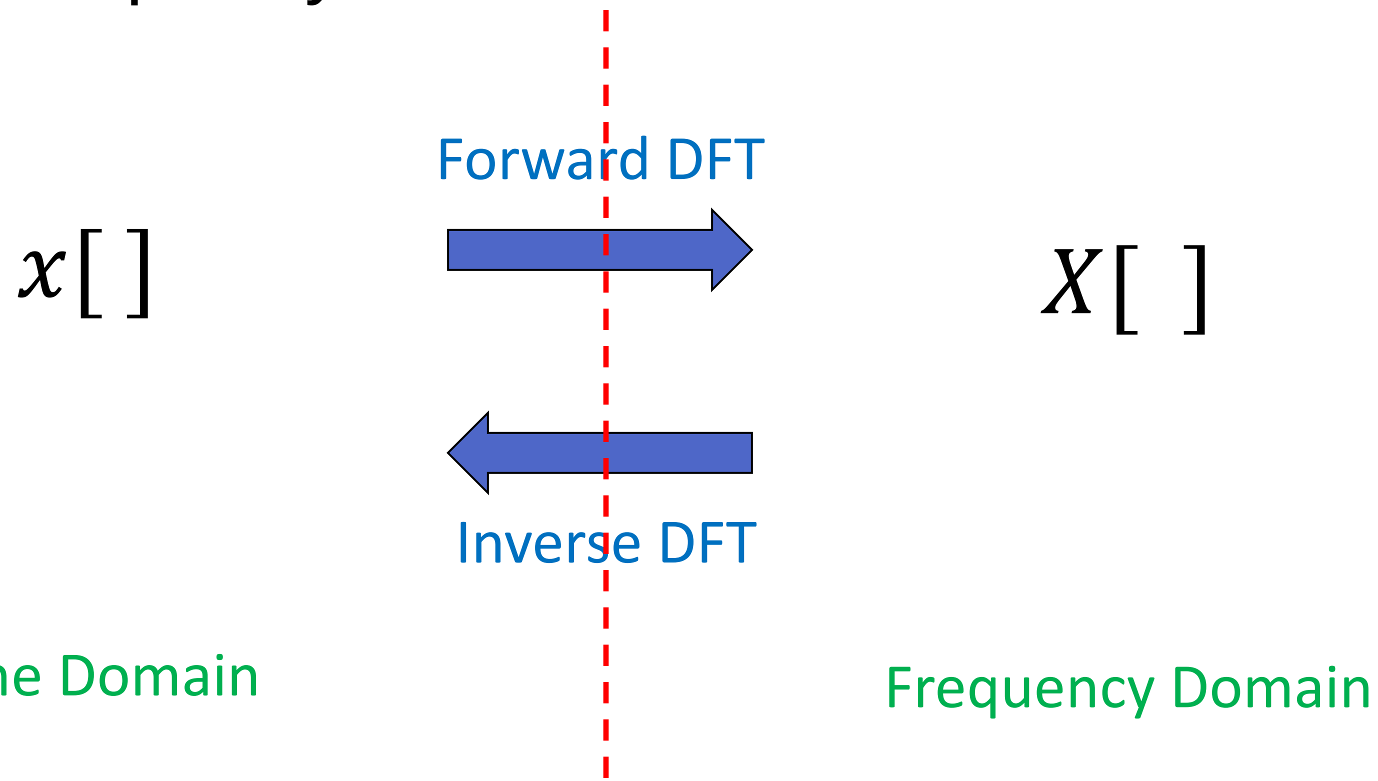
Applications of the DFT

Today's Topics

- DFT Review
- Spectral Analysis of Signals
  - Frequency resolution and DFT length
  - DFT and Averaging
  - DFT and windowing
- Time and Frequency Domain Relationships

Discrete Fourier Transform

- The DFT transforms a time domain signal into the frequency domain



Discrete Fourier Transform

- N samples in the time domain produce $N/2 + 1$ samples in the frequency domain

Forward DFT

$x[\,]$

$X[\,]$

N-Samples in the Time Domain

Inverse DFT

N/2 +1 Samples In the Frequency Domain

Time Domain

Frequency Domain

BASIS Functions for the DFT

- The DFT maps the input signal into a COS and SINE using the basis functions

$$c_k[i] = \cos(2\pi ki/N)$$

$$s_k[i] = \sin(2\pi ki/N)$$

- These represent COS and SINE functions that have a frequency of k/N

- The function will complete k cycles in N samples

BASIS Functions for the DFT

- The BASIS functions for the DFT are

$$c_k[i] = \cos(2\pi ki/N)$$

$$s_k[i] = \sin(2\pi ki/N)$$

- i goes from 0 to N-1 and represents the time domain

- k goes from 0 to N/2 and represents the frequency

But How do We Get $X[k]$?

- We *correlate* the input sequence with each COS and SINE wave at $N/2+1$ frequencies

$$Re(X[k]) = \sum_{i=0}^{N-1} x[i]\cos(2\pi ki/N)$$

$$Im(X[k]) = \sum_{i=0}^{N-1} x[i]\sin(2\pi ki/N)$$

Multiply the input sequence by N samples of the cosine and sine signals for each frequency k

Discrete Fourier Transform

- The real part of the frequency domain signal are the COSINE amplitudes. Imaginary part are the SINE amplitudes

Frequency Domain

$$X[\ ]$$

N/2 +1 Samples
In the Frequency
Domain

$$Re[X]$$

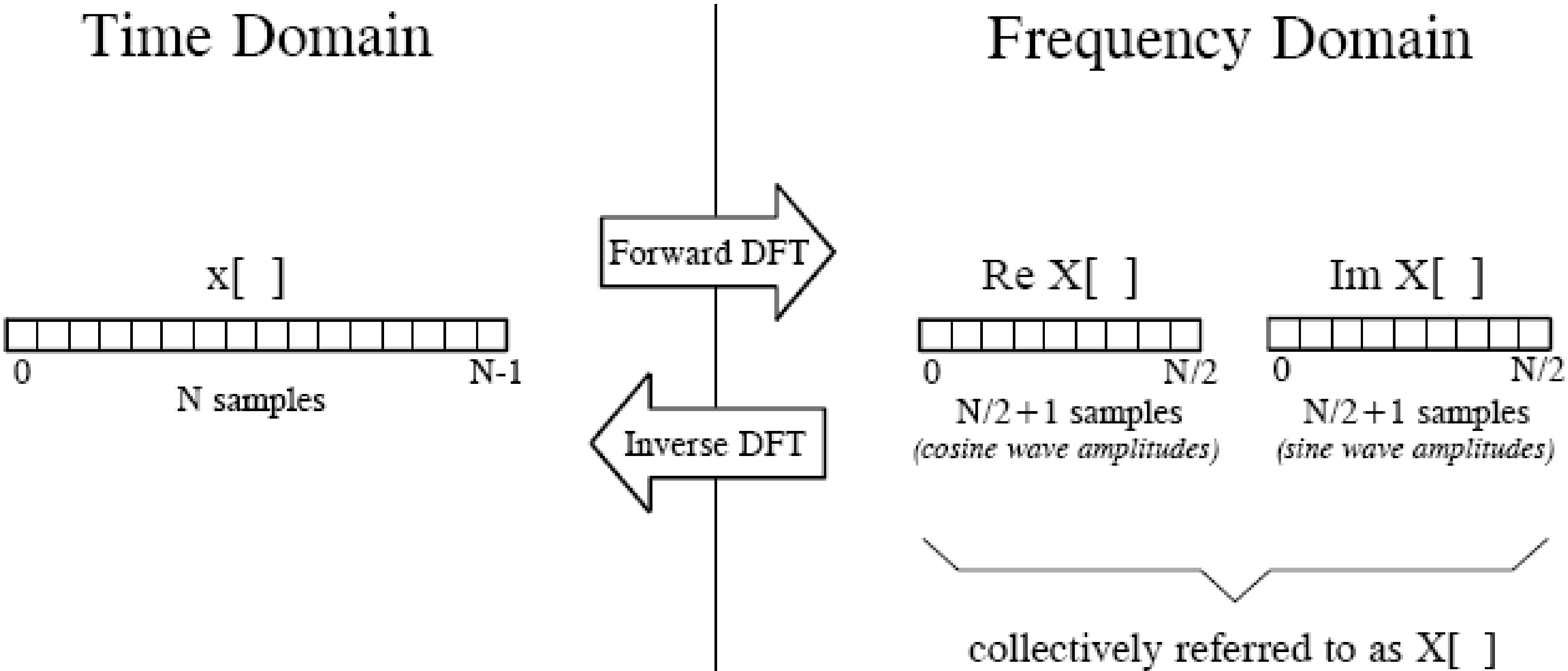
$$Im[X]$$

Amplitude of the
COSINE waves

Amplitudes of the
SINE waves

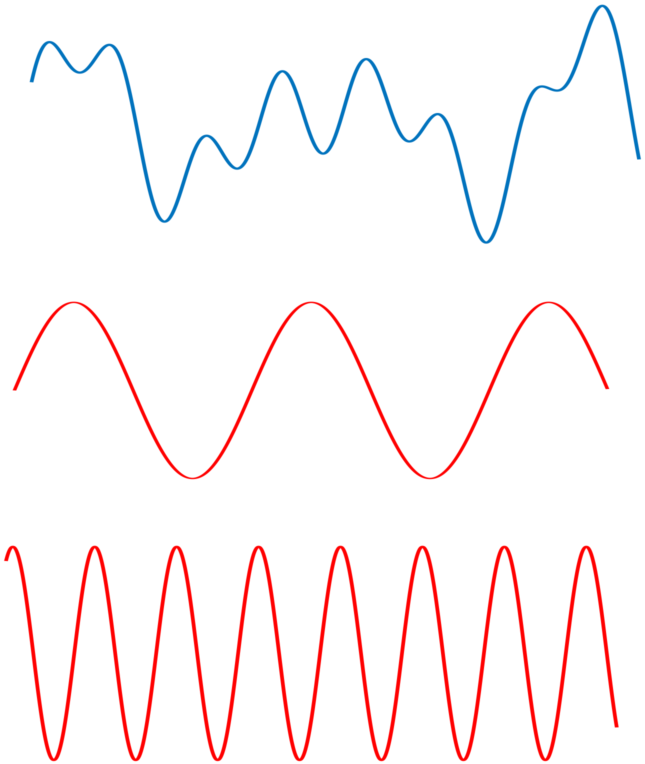
Real DFT: Time to Frequency Domain Transform

- Frequency Domain refers to the amplitude of cosines/sines



Correlating with Sinusoids

Input Signal



The DFT values are essentially the result of correlations

Correlate the input signal with sine and cosine waves of N/2 frequencies from 0 to $f_s/2$

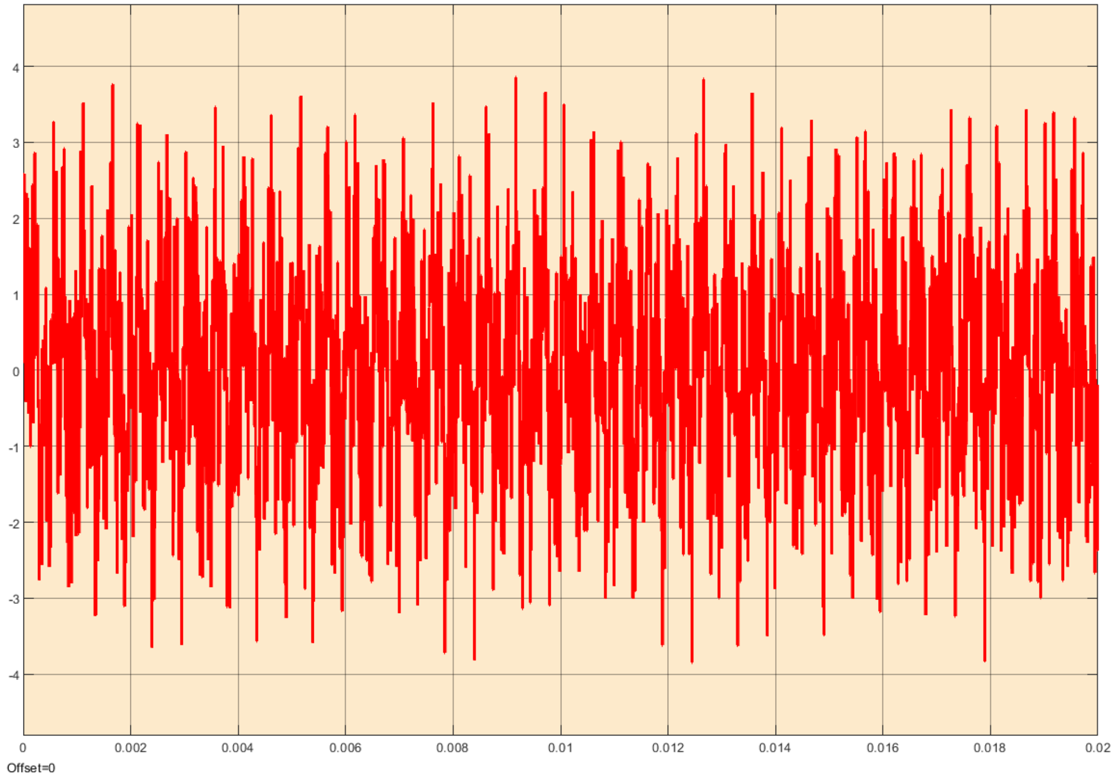
How much of each frequency is contained in the signal?

Spectral Analysis of Signals

- Information about many physical systems is encoded in the frequency and amplitude of the component sinusoids.
  - E.g. speech, etc.
- The time domain signal may be buried in noise
  - Difficult to visually determine the features of the waveform

SIMULINK Example

- Signal in the time domain

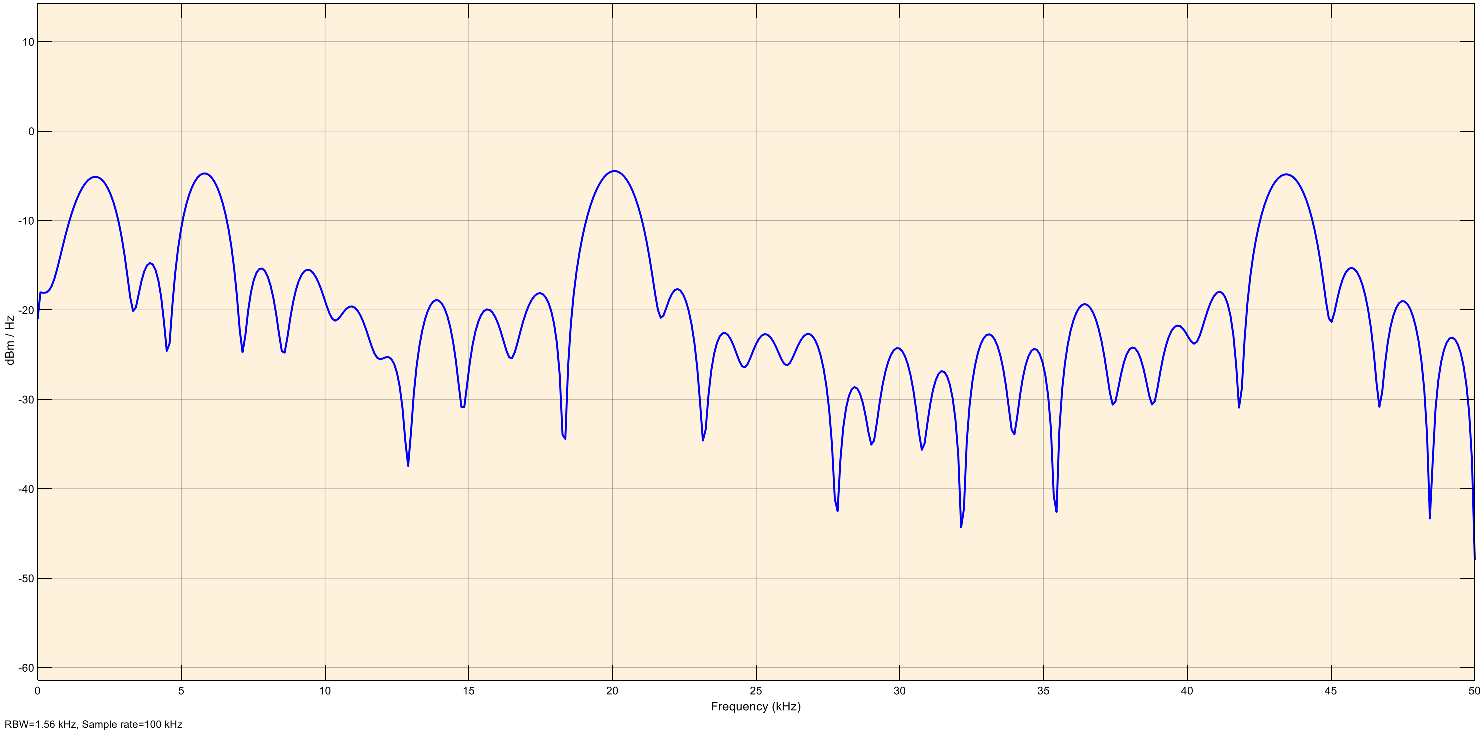


Difficult to see clear structure in the signal

DFT_Demo.slx

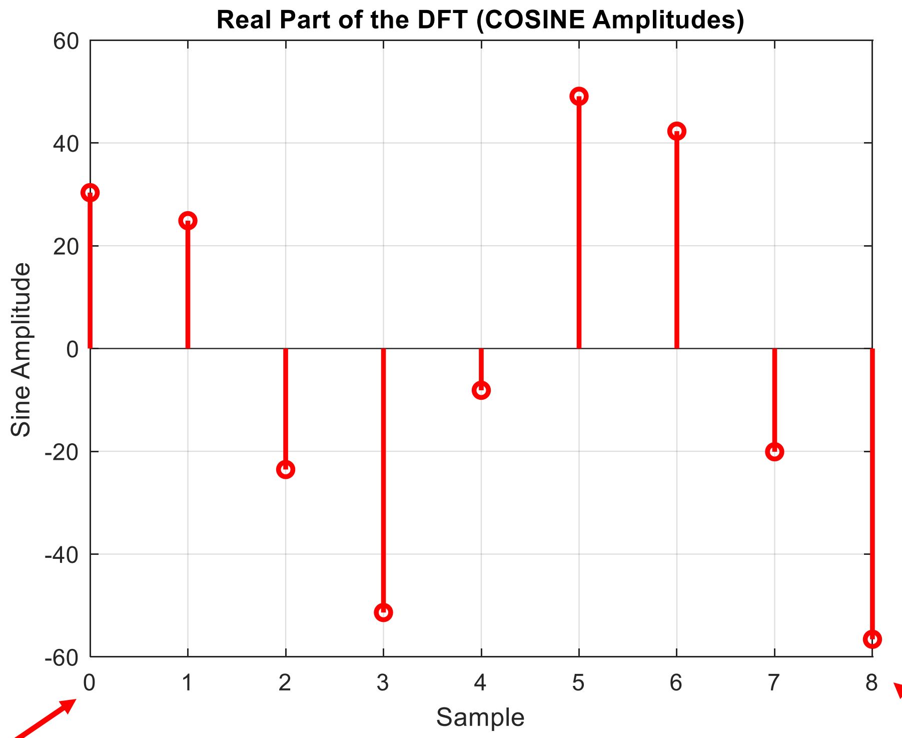
SIMULINK Example

- In the frequency domain some signal structure is forming – Better resolution may help distinguish signals



What is the Frequency Resolution of my DFT?

- If a signal has N samples then the DFT will create $N/2+1$ samples



The frequency range is from 0 to $f_s/2$

0

$N/2+1$ *samples*

$f_s/2$

What is the Frequency Resolution of my DFT?

- The frequency resolution is then

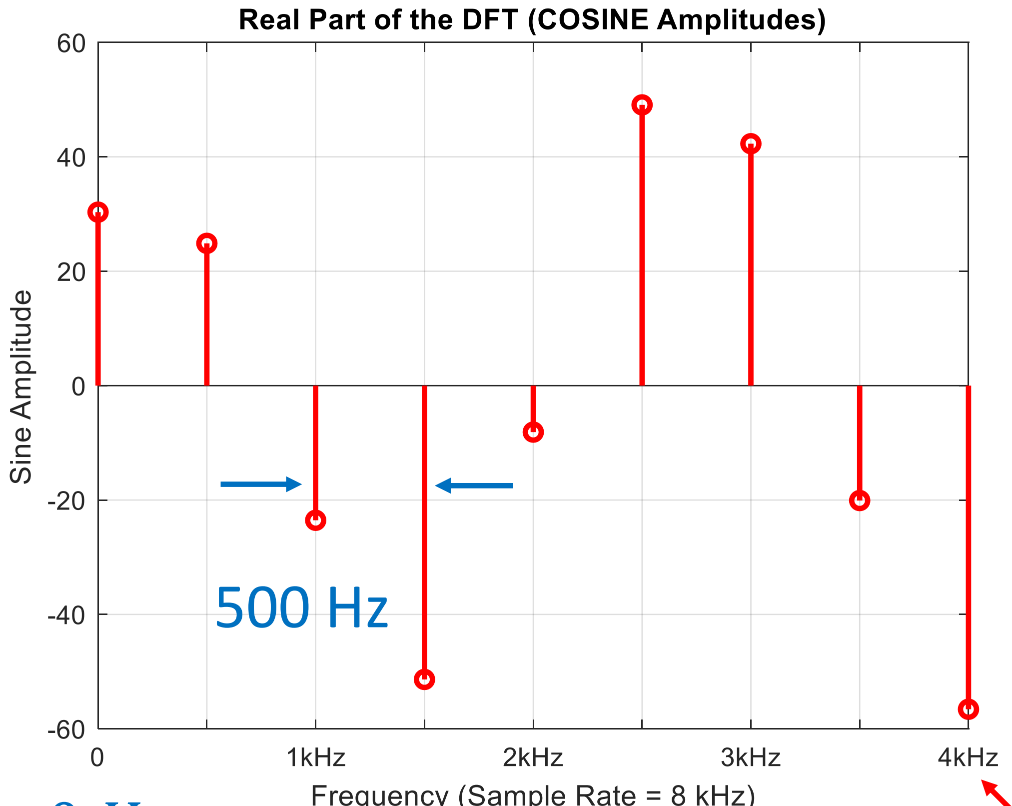
$$f_{res} = \frac{f_s/2}{N/2} = \frac{f_s}{N}$$

- Example – Sampling at 8 kHz and using 16 samples

$$f_{res} = \frac{8kHz}{16} = 500\ \mathrm{Hz}$$

Frequency Domain Independent Variable

- Frequency resolution of 500 Hz



$0\ Hz$

$f_s/2 = 4000\ Hz$

How Can I Improve Resolution?

- Use more points from the sample

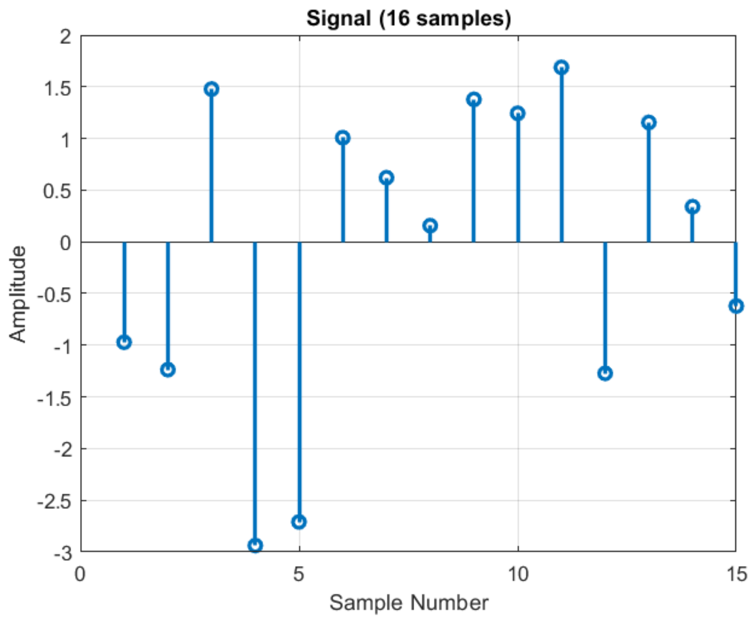
- Example – Sampling at 8 kHz and using 64 samples of the signal

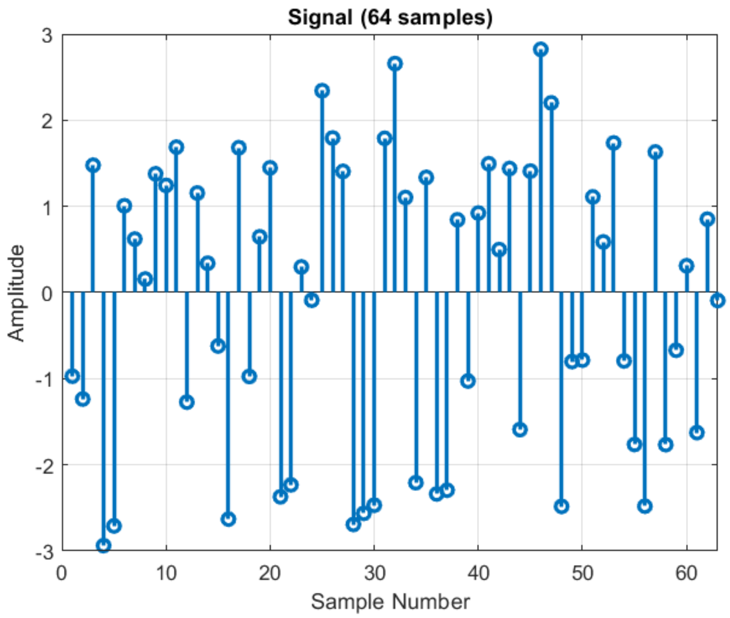
$$f_{res} = \frac{f_s/2}{N/2}$$

$$f_{res} = \frac{8kHz/2}{64/2} = 125\text{ Hz}$$

Improving Resolution

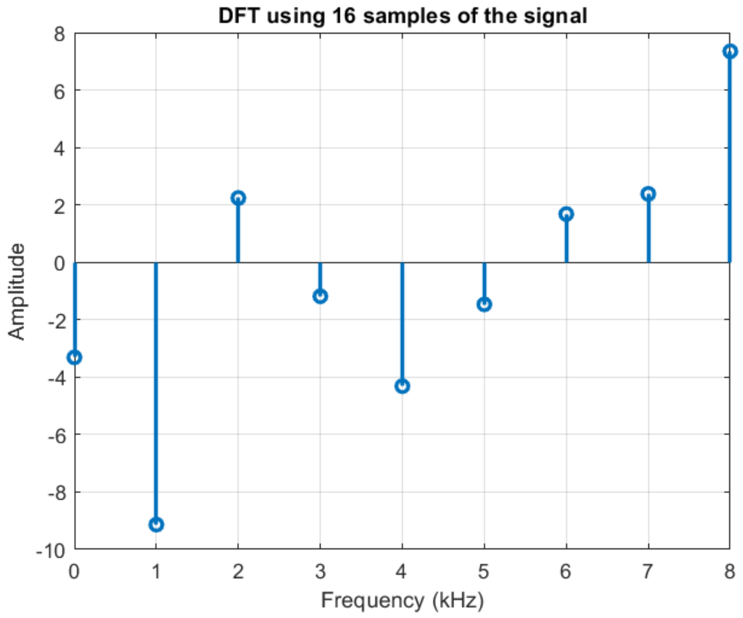
- To improve frequency resolution, take more samples of the input and take the DFT

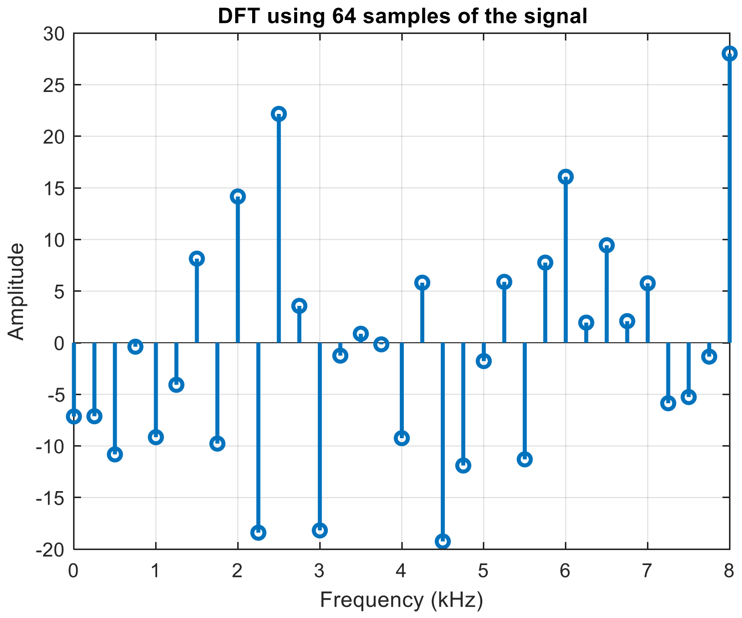




Improving Resolution

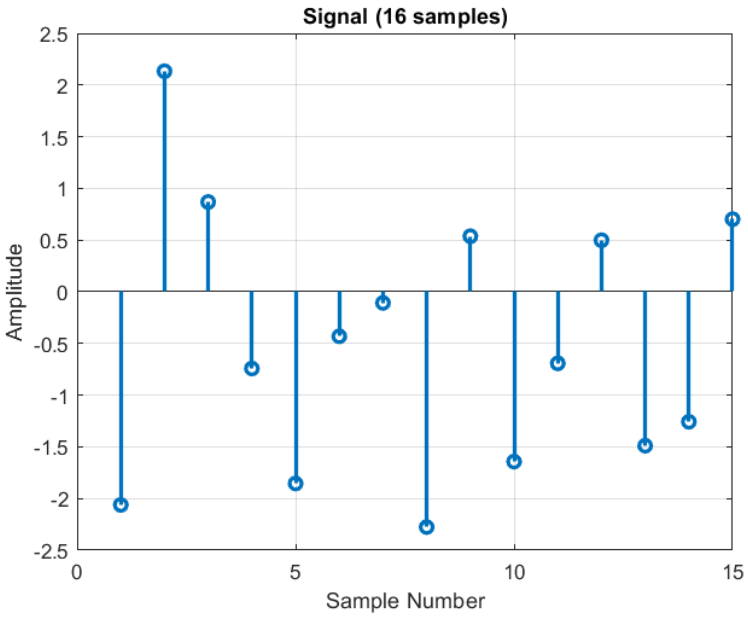
- Frequency resolution improved from 500 Hz to 125 Hz

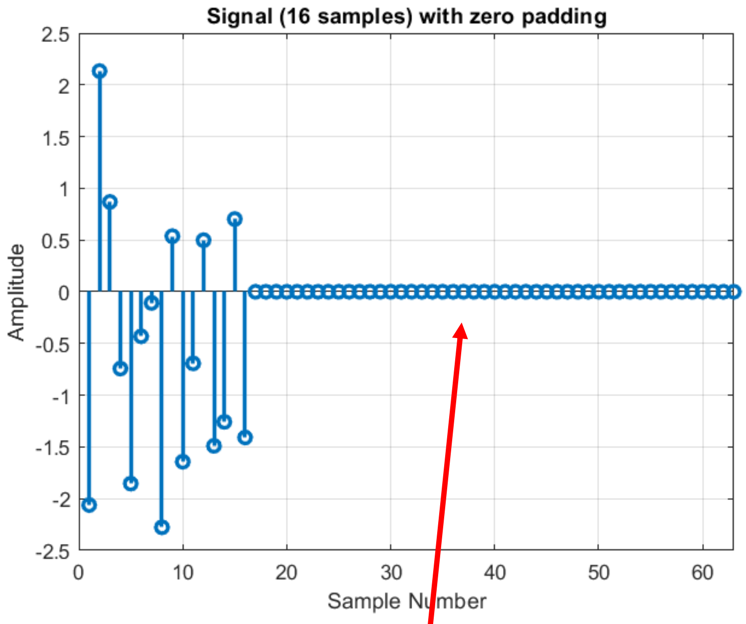




What If I Don't Have More Samples?

- I can extend the length of the DFT by padding my signal with zeros

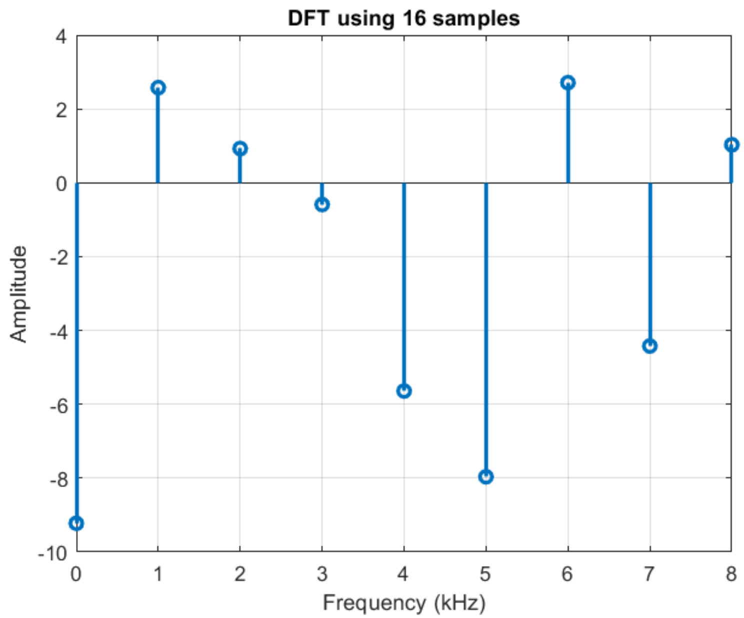


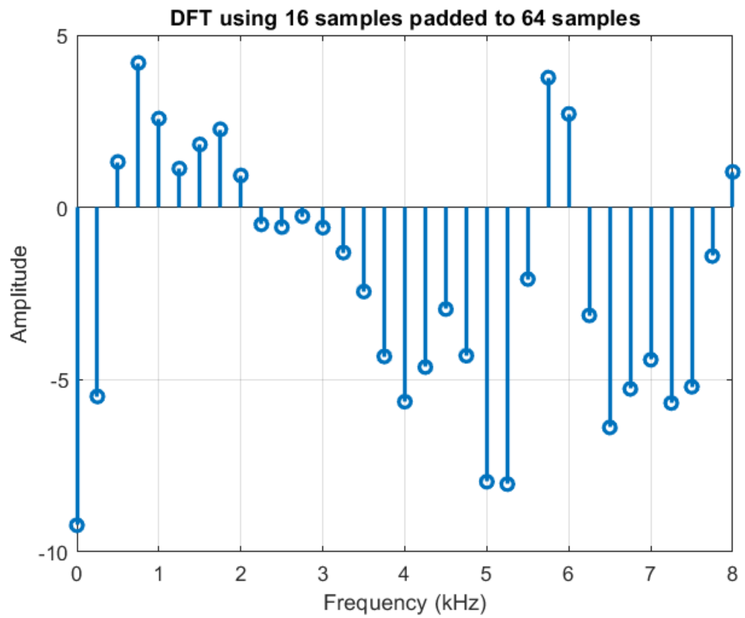


48 zeros added to the end of the sequence

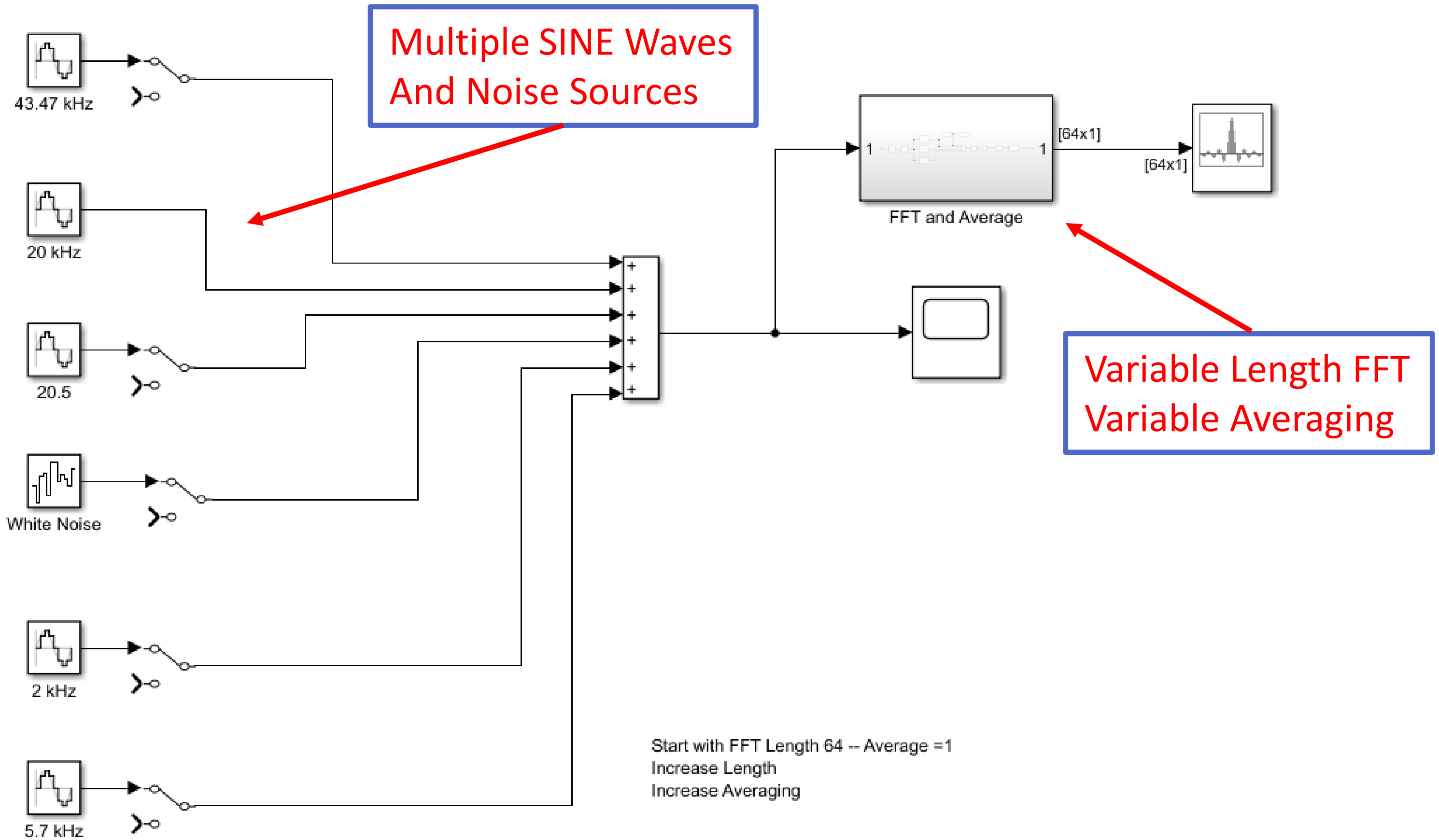
DFT After Zero Padding

- Padding with zeros improved the frequency resolution of the DFT





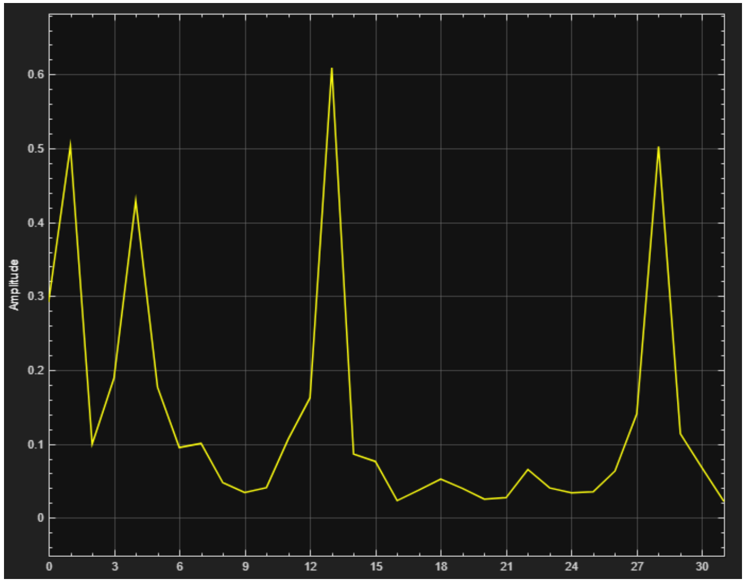
SIMULINK Example
Improving Frequency Resolution

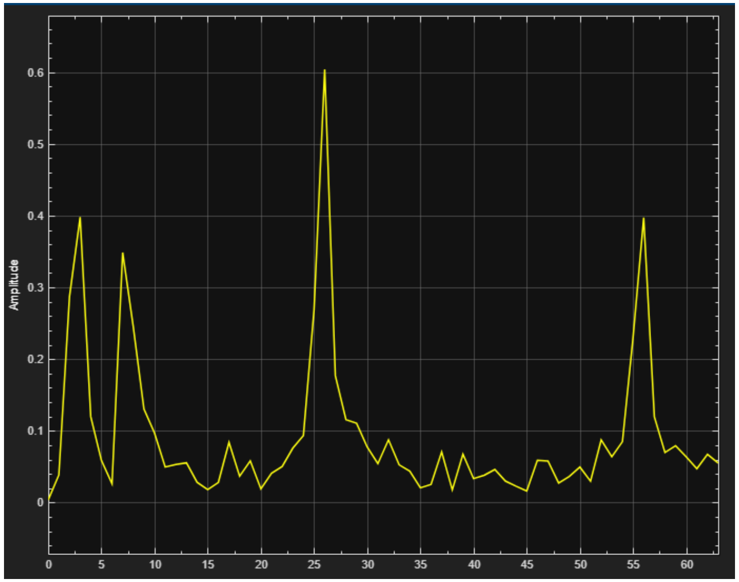


DFT_Demo.slx

SIMULINK Example
Improving Frequency Resolution

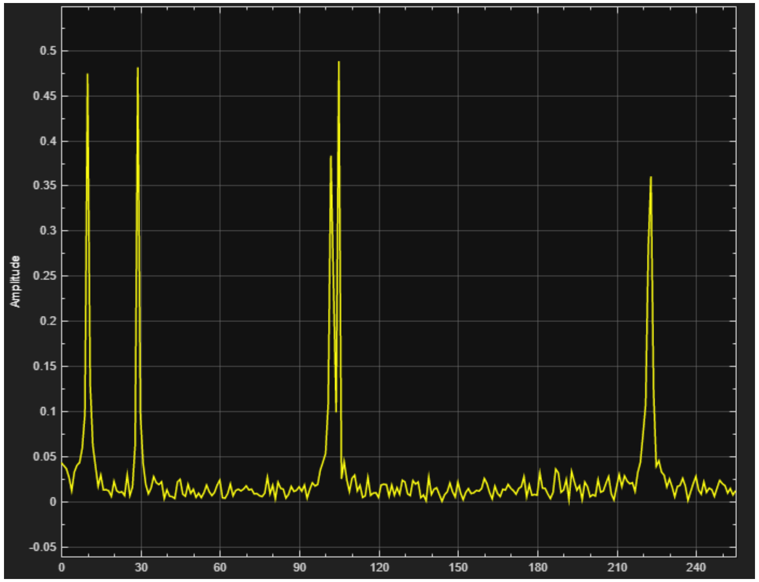
- 64 and 128 samples used in the DFT

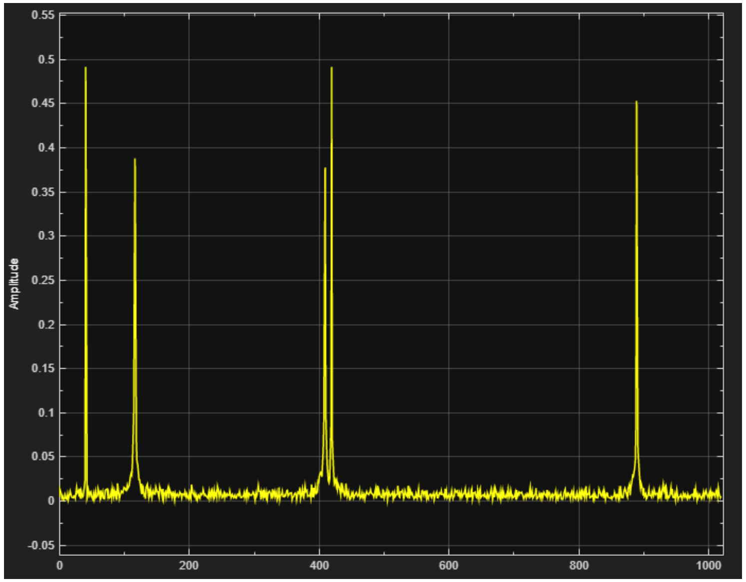




DFT Length

- 512 and 2048 samples used in the DFT
- Has the noise level improved?





Decreasing Noise Levels

- How have we decreased noise on signals in the time domain?

- Averaging – Recall the noise reduction from averaging

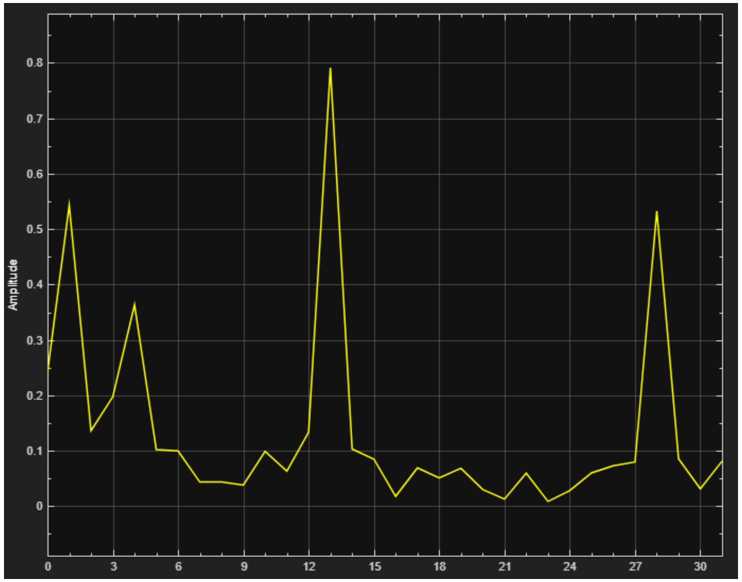
- For N samples

$$\sigma_{ave} = \frac{\sigma_{signal}}{\sqrt{N}}$$

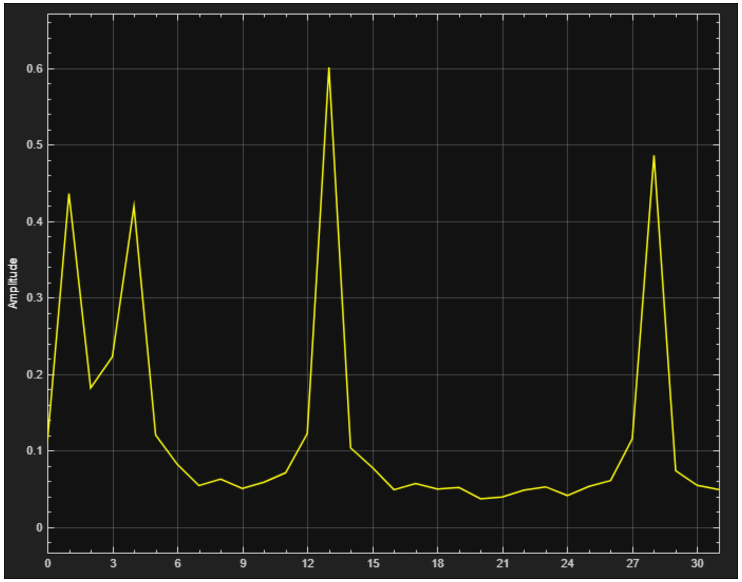
Decreasing Noise Levels

- What if we take multiple DFT outputs and average their magnitudes?

64 Samples, 1 Average

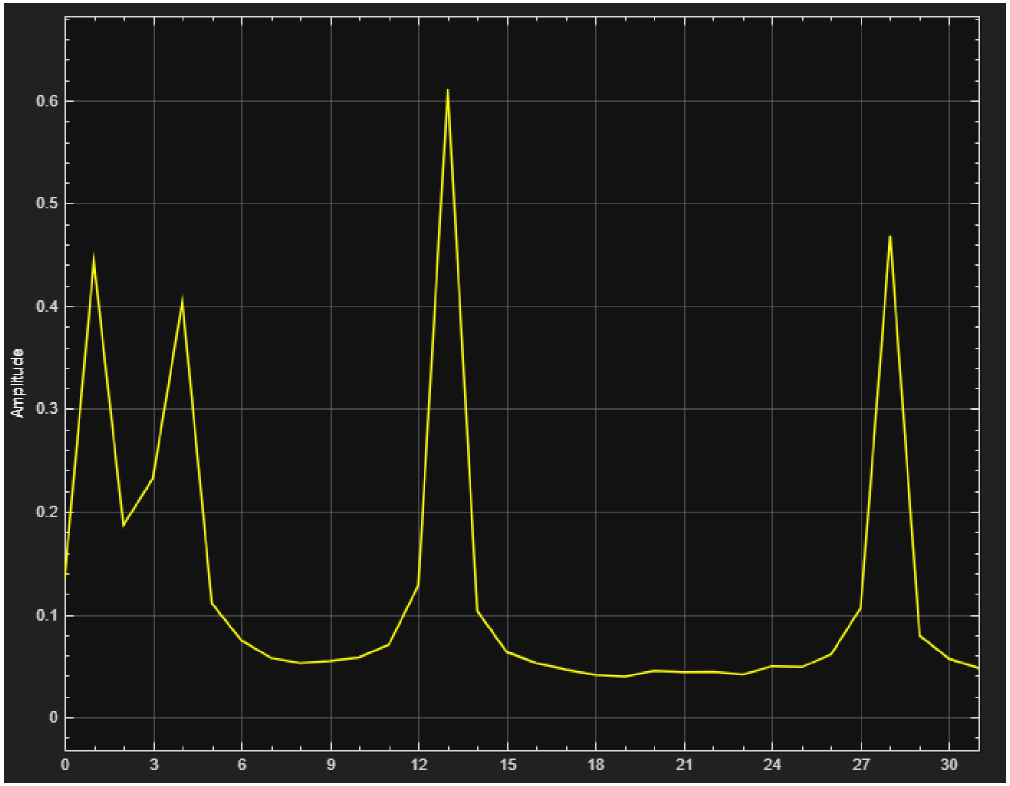


64 Samples, 10 Averages



Decreasing Noise Levels

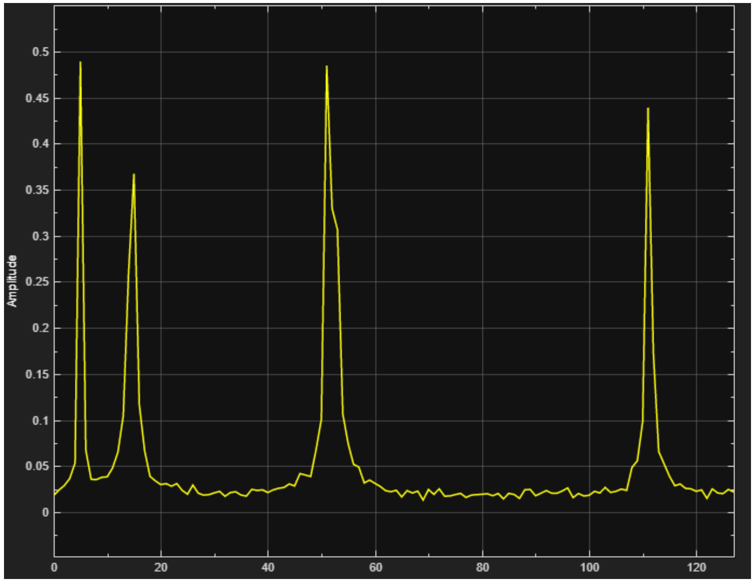
- 64 Samples – 100 averages



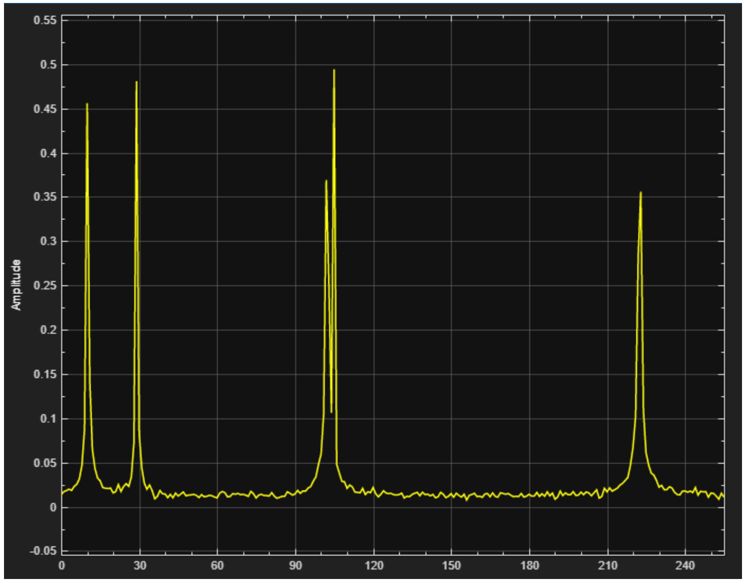
Combining Sample Size and Averaging

- Increase sample size and number of averages

256 Samples, 10 Average



512 Samples, 100 Averages



Alternative Methods for Averaging Data

- Use a long DFT to get more data.
  - Improves frequency resolution but does not decrease noise level.

- Break the input into smaller sample segments, and average DFT outputs.
  - Gives low noise and defined resolution. Computationally fast algorithm

- Use a long DFT to get more data
  - Then smooth the DFT using a low pass filter
  - This improves frequency resolution and regains lower noise
  - Difficult to calculate

More on Resolution and Windowing

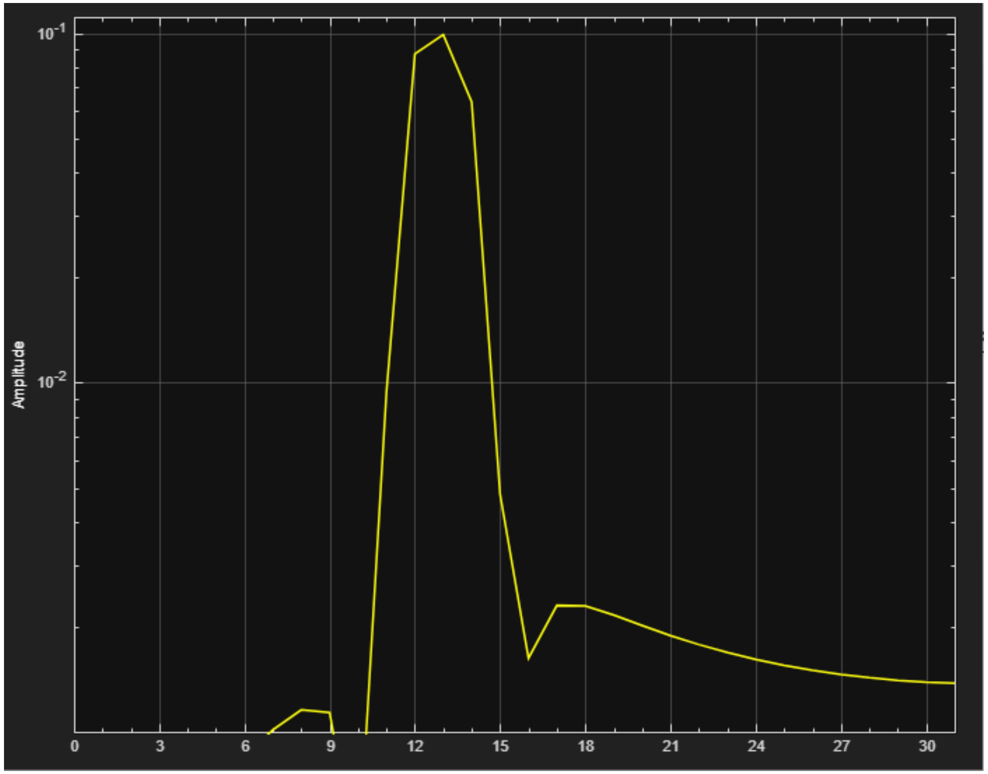
- I have two signals that are very close in frequency – 20 kHz and 20.5 kHz

- I am sampling at 100 kHz

- What does my DFT look like with 64 samples of the signal?

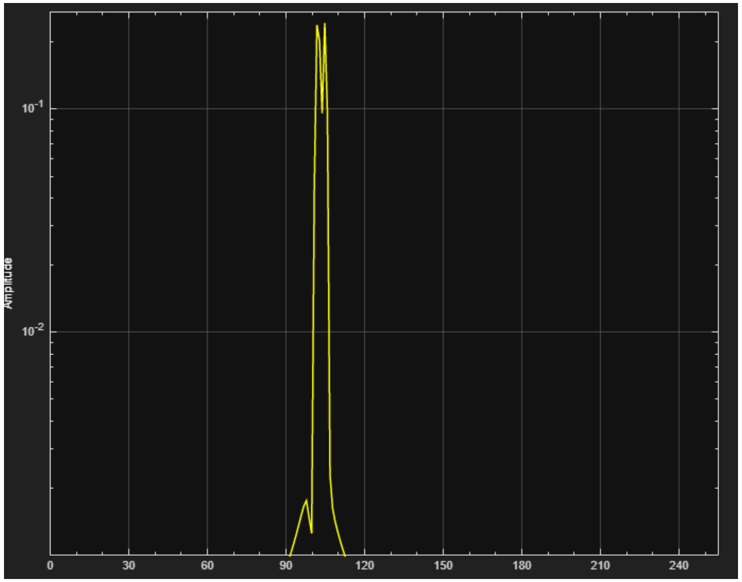
More on Resolution and Windowing

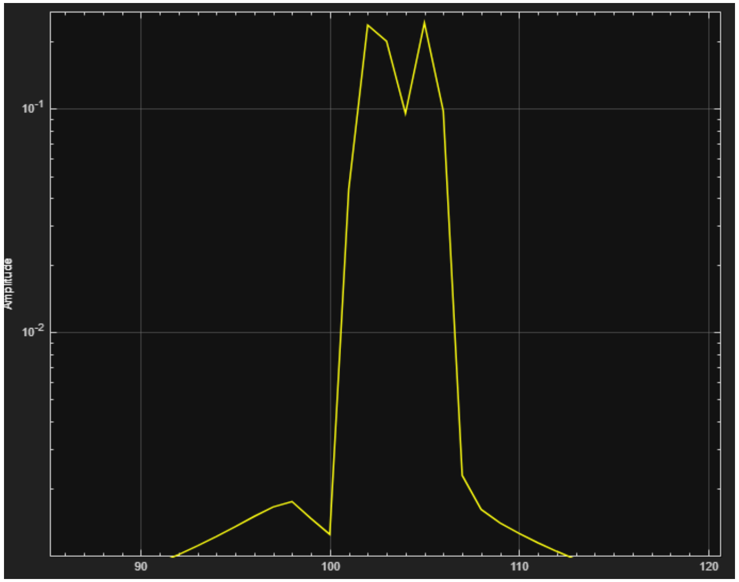
- With 64 samples the DFT doesn't show a distinct difference between the two tones



Try Increasing N to 512

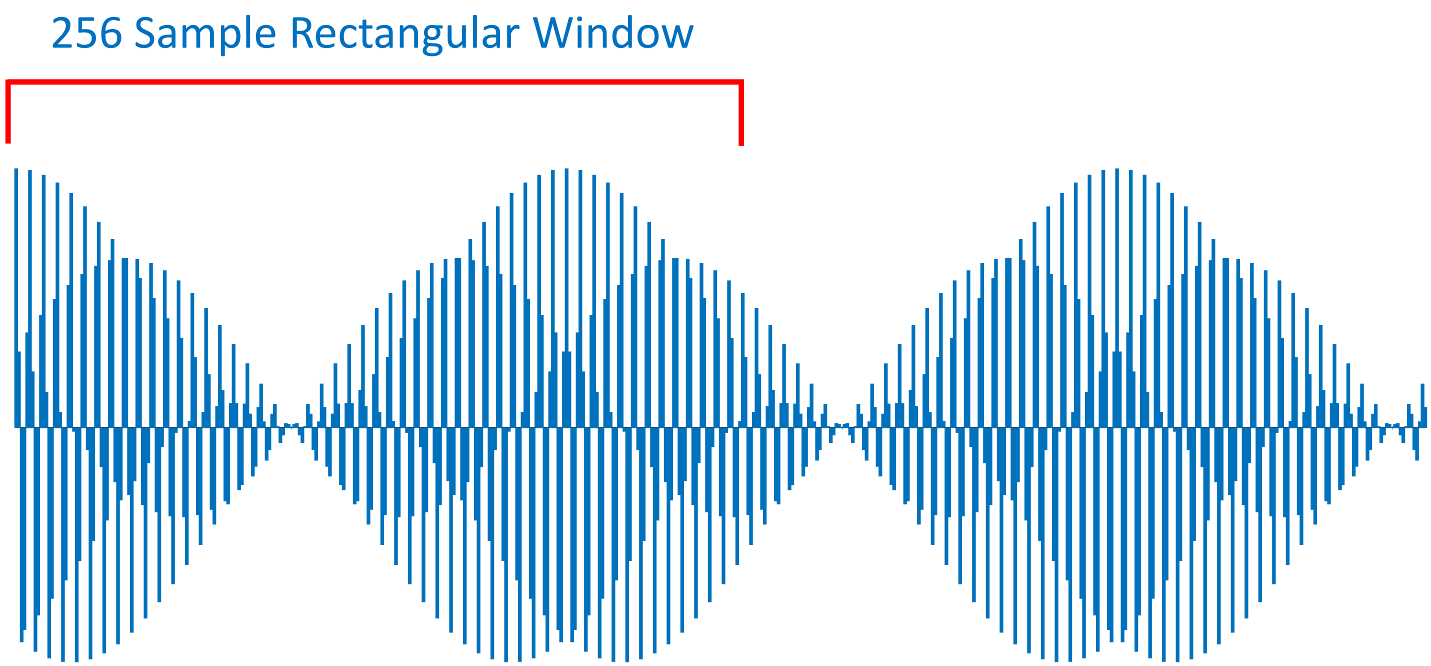
- Increasing the number of samples allows me to resolve the two signals





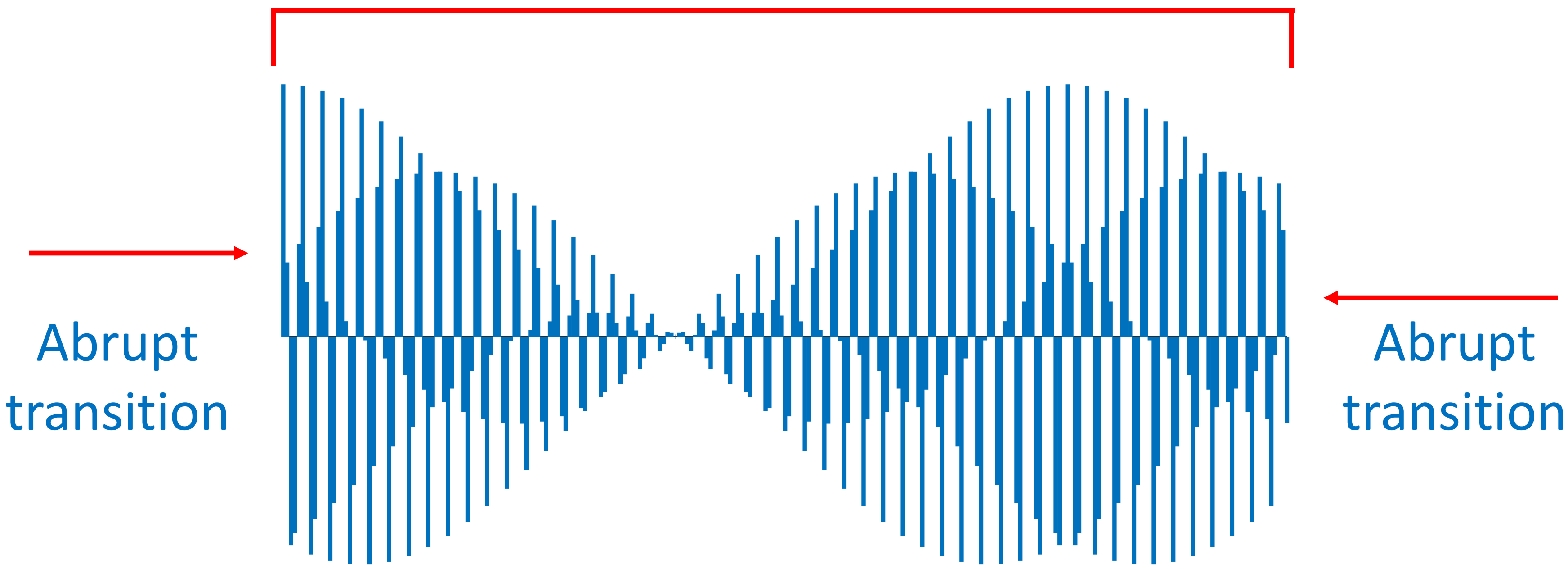
What is Windowing?

- When I take my set of N samples I grab a set of N samples without modification – Rectangular window



Rectangular Window

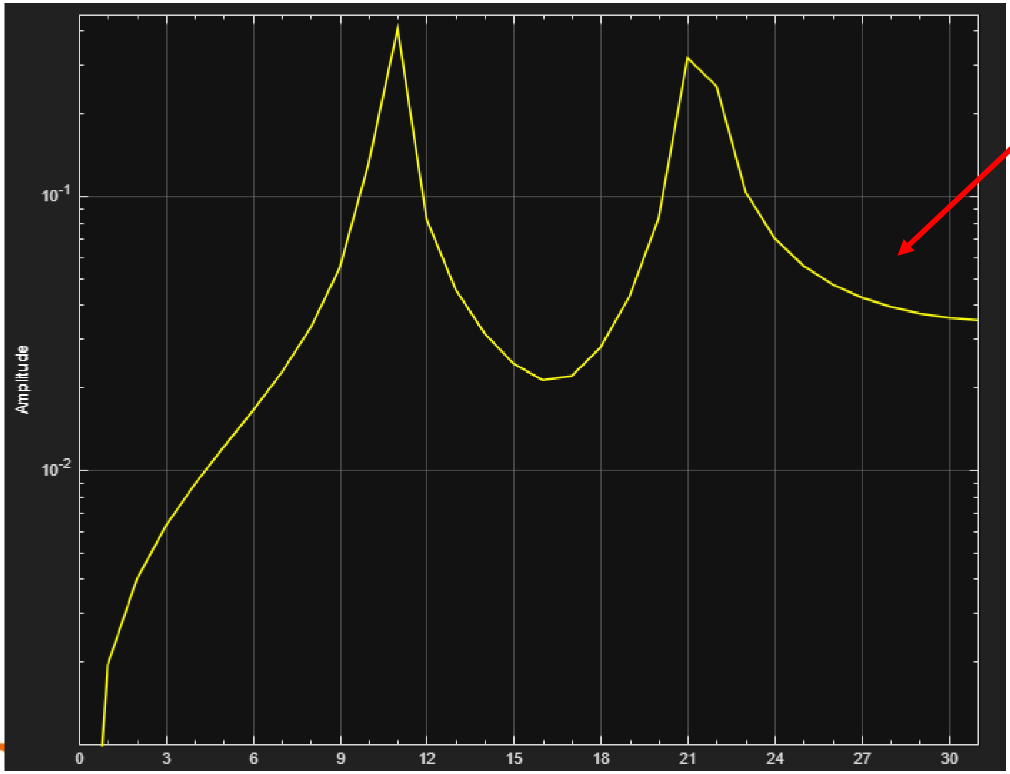
- The rectangular window has sharp transitions at the end of the window
- Can cause “leakage” in the frequency domain



Rectangular Window

- The Peak of the tone is broad and there are “tails” in the spectrum

Tails



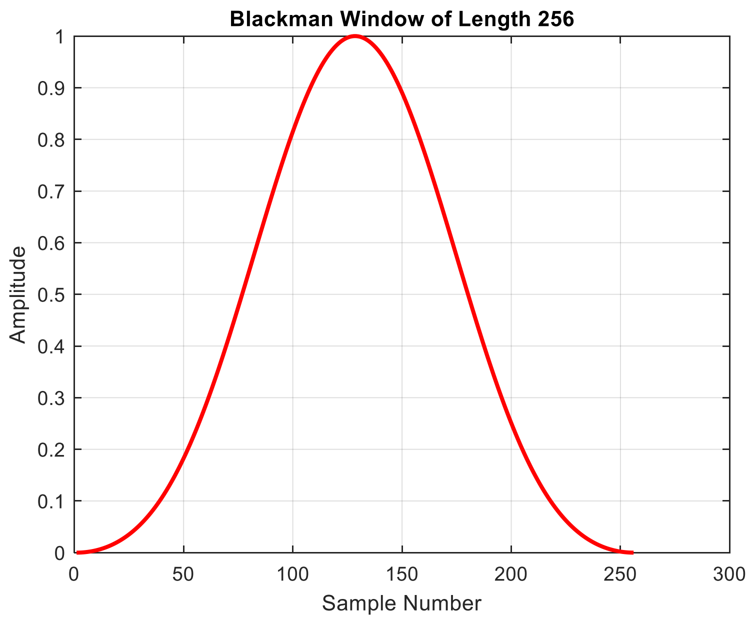
Windowing

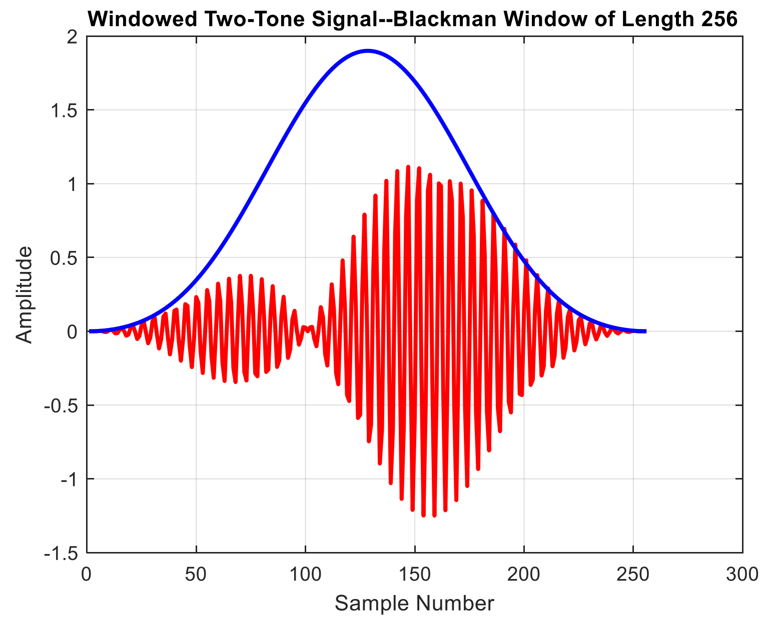
- A window can be applied to reduce the impacts of the abrupt transition on the end of the signal

- Common windows are:
  - Hamming Window
  - Blackman Window

Blackman Window

- The window multiplies the sample values in time and reduces the abrupt transitions

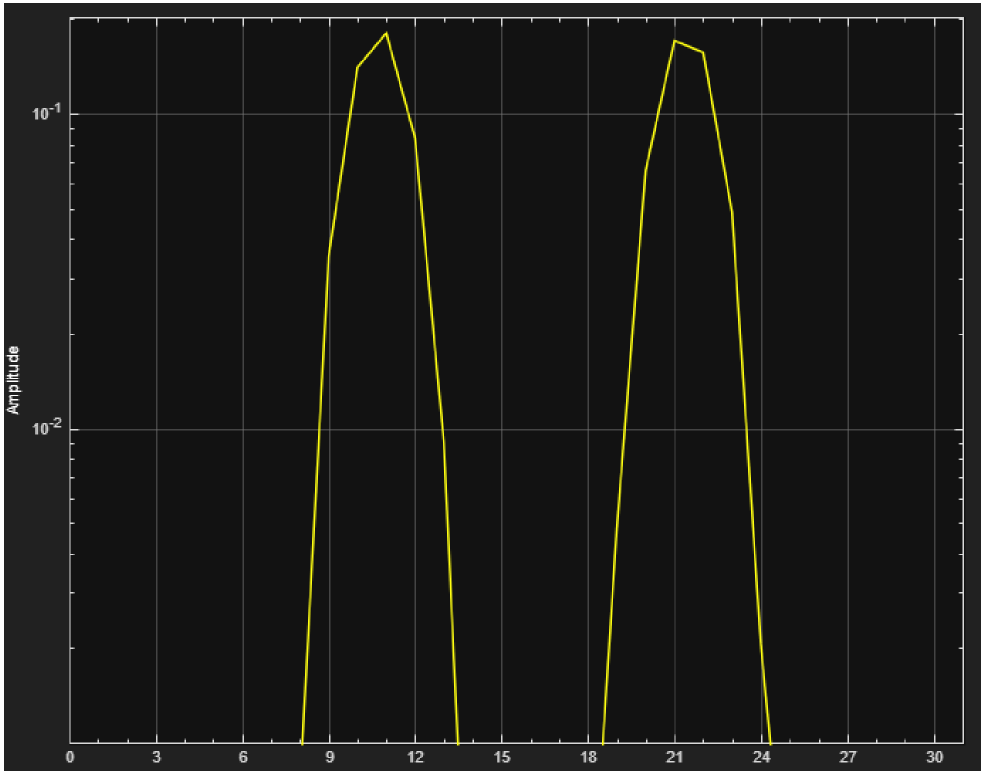




$$w[n] = 0.42 - 0.5\cos\left(\frac{2\pi n}{N}\right) + 0.08\cos\left(\frac{4\pi n}{N}\right)$$

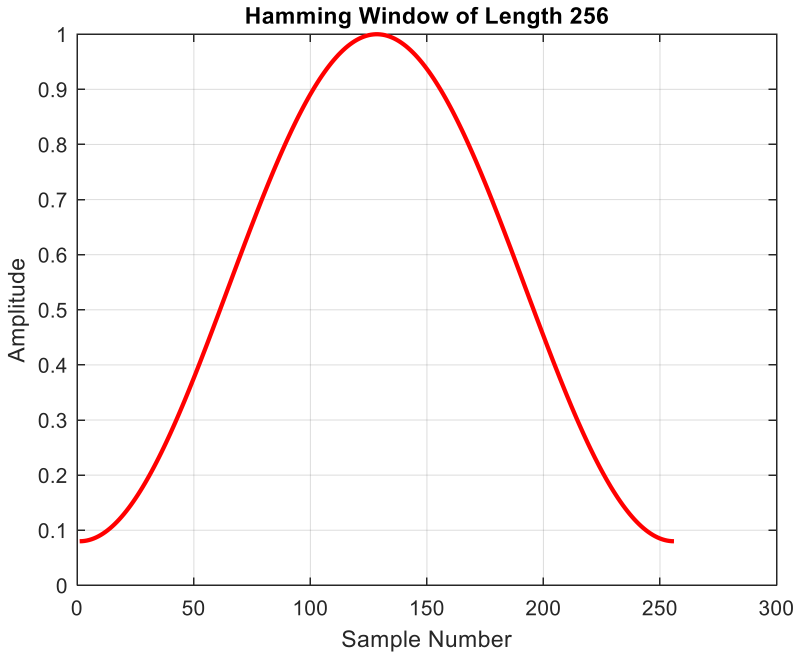
Blackman Window

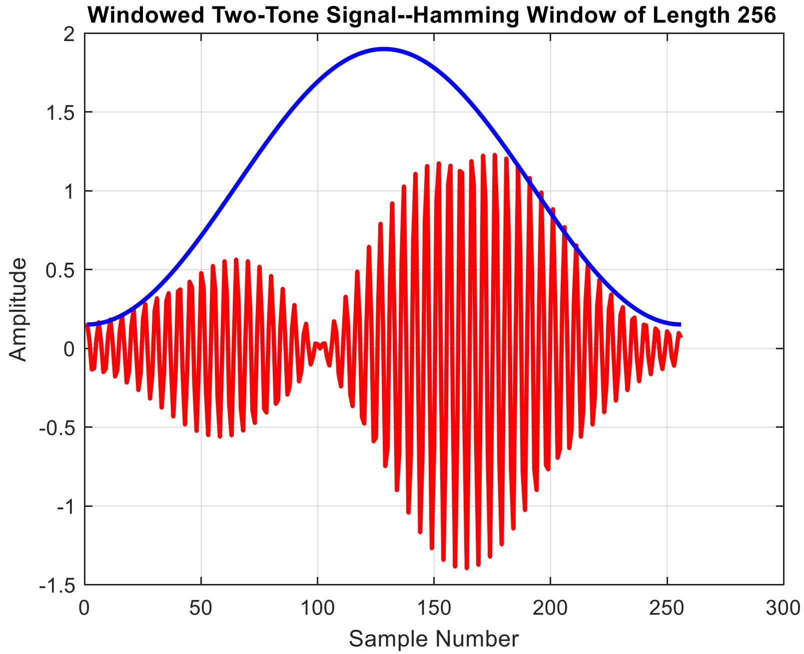
- The Blackman window broadens the peak but reduces the side lobes significantly



Hamming Window

- The window multiplies the sample values in time and reduces the abrupt transitions

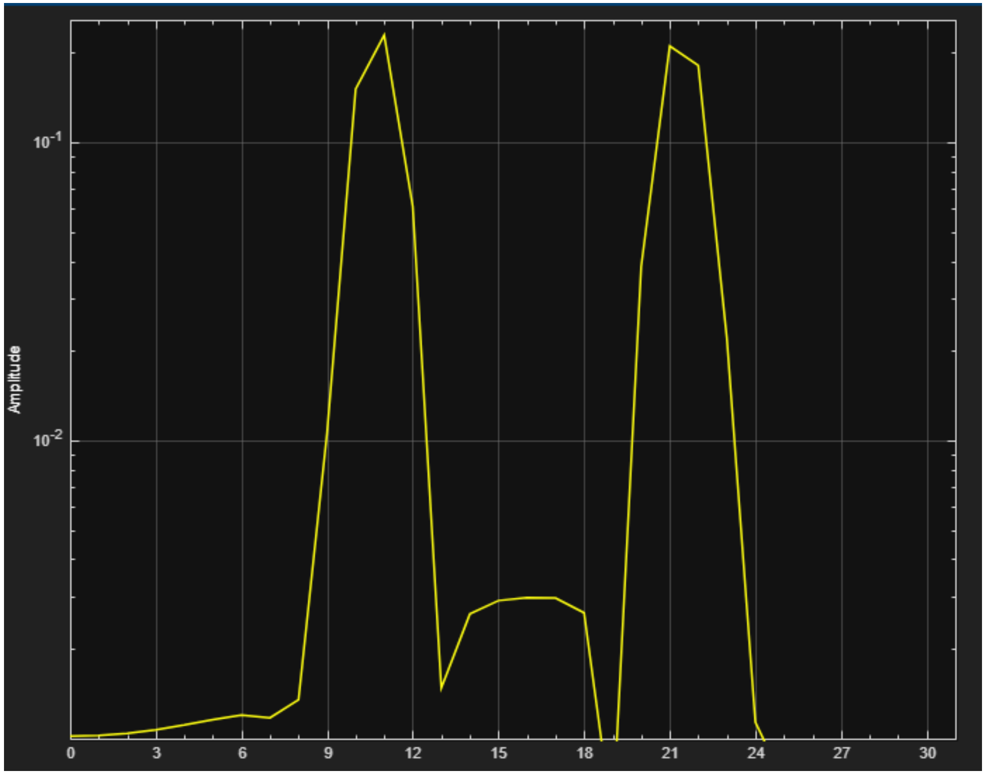




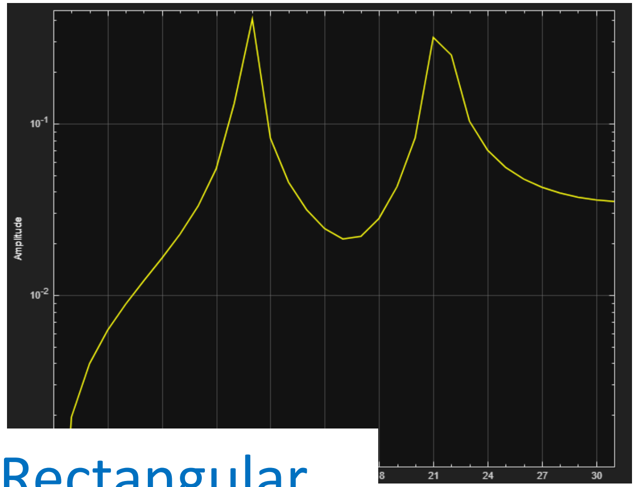
$$w[n] = 0.54 - 0.46\cos(2\pi\frac{n}{N})$$

Hamming Window

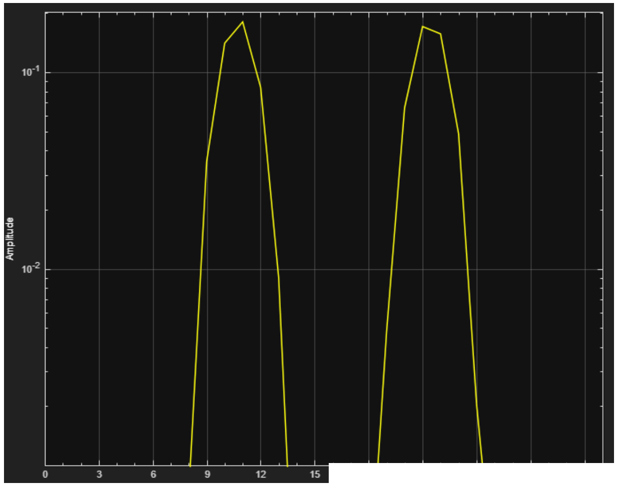
- Is in between the rectangular window and the Blackman window



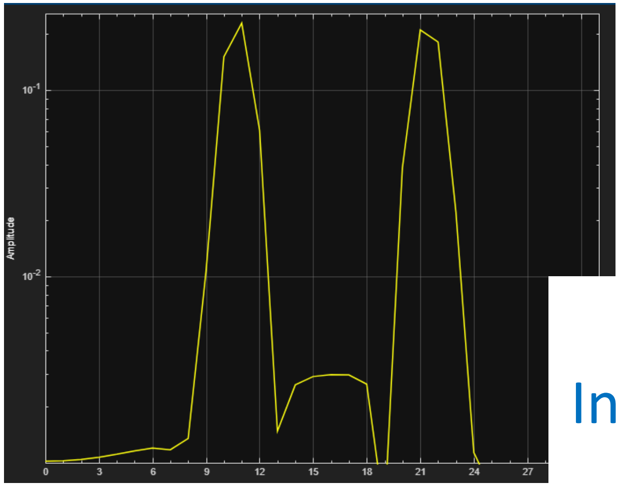
Window Comparison



[Rectangular]
Narrowest Peak
Highest Tails



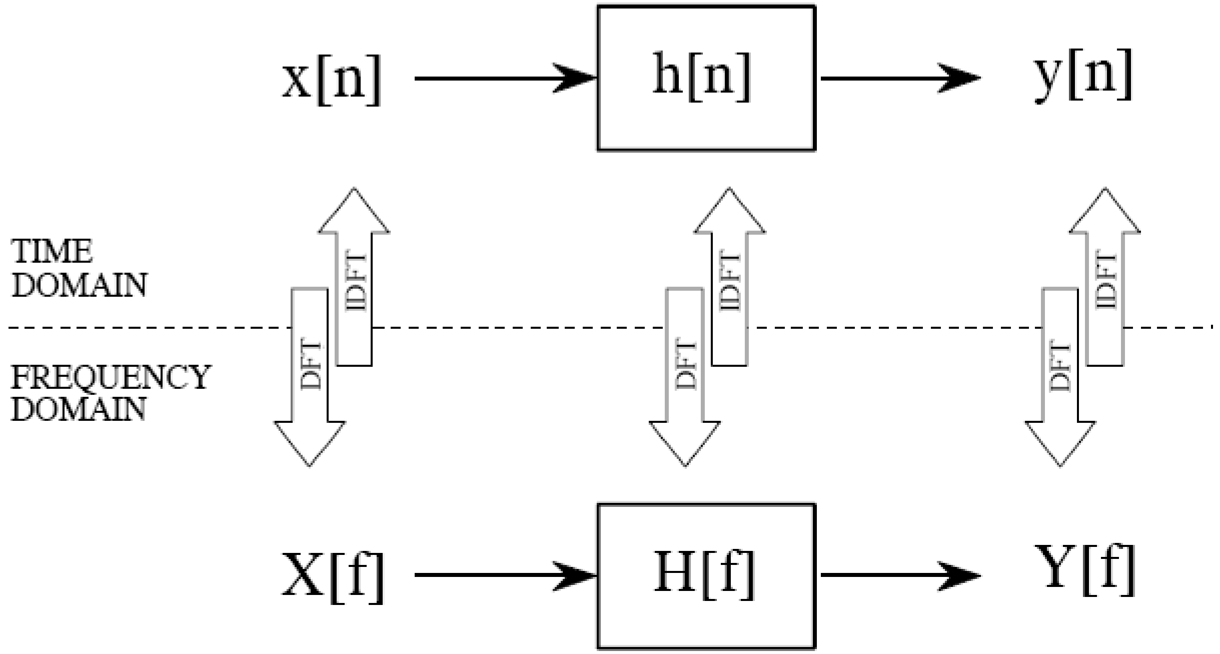
[Blackman]
Widest Peak
Lowest Tails



[Hamming]
In between Hamming
and Blackman

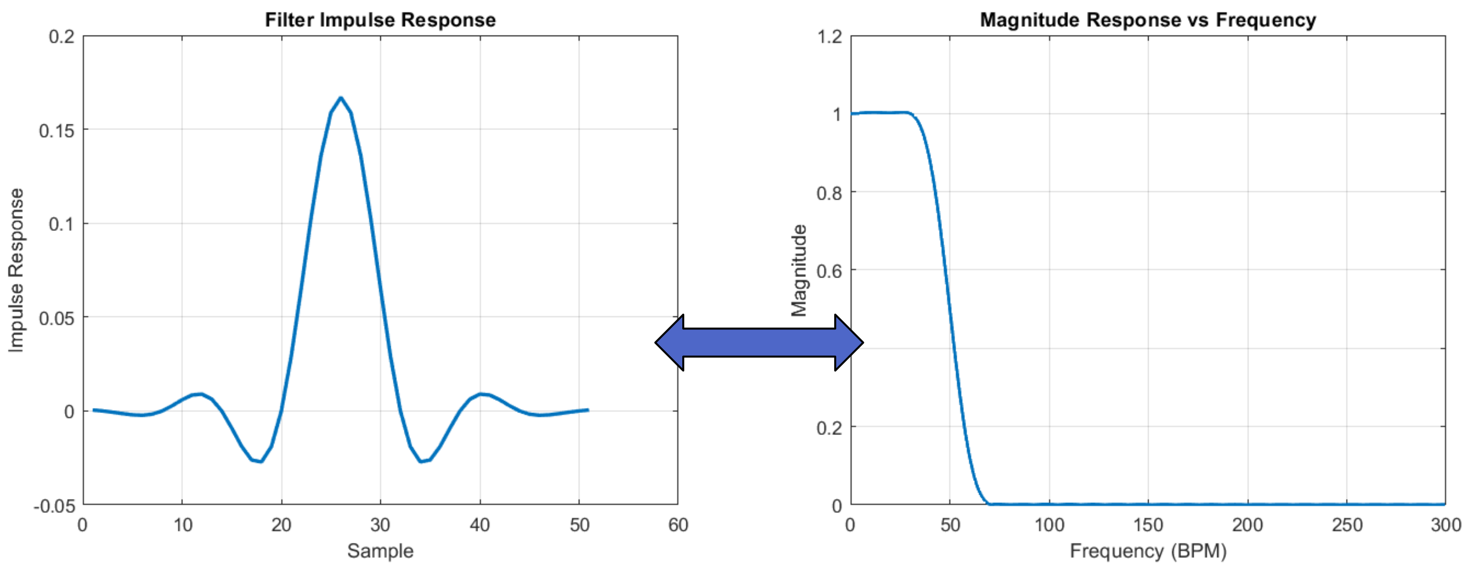
Time and Frequency Domain Relationships

- There is a correspondence between time domain and frequency domain



Time and Frequency Domain Relationships

- The impulse response in the time domain corresponds to the frequency response in the frequency domain

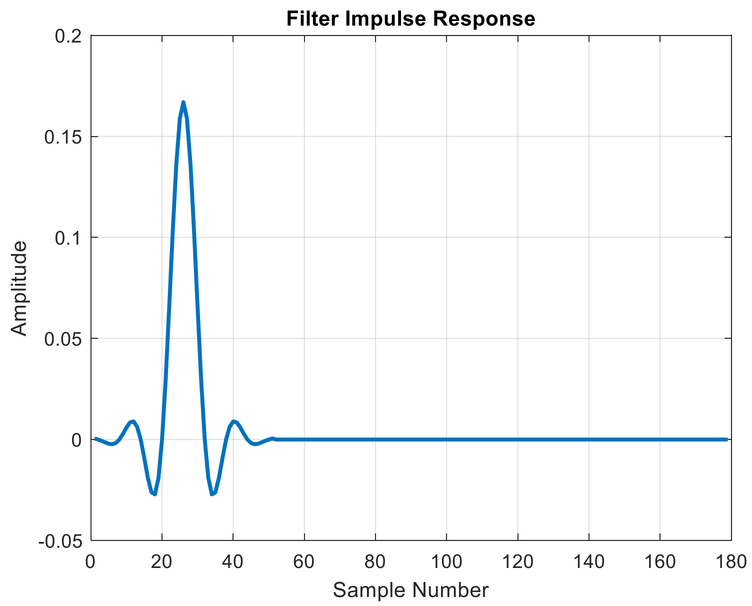


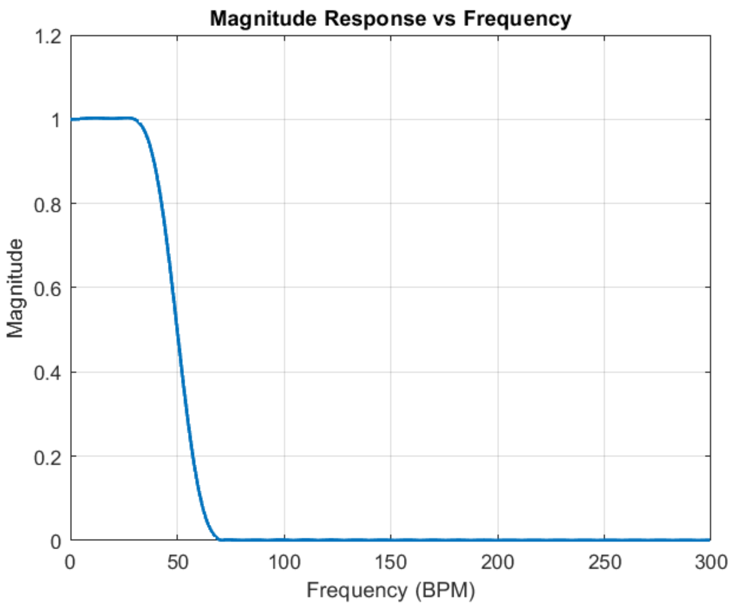
Time Domain   Frequency Domain

43

Impulse Response and Frequency Response

- By padding the impulse response with zeros we can get better frequency resolution for the frequency response.

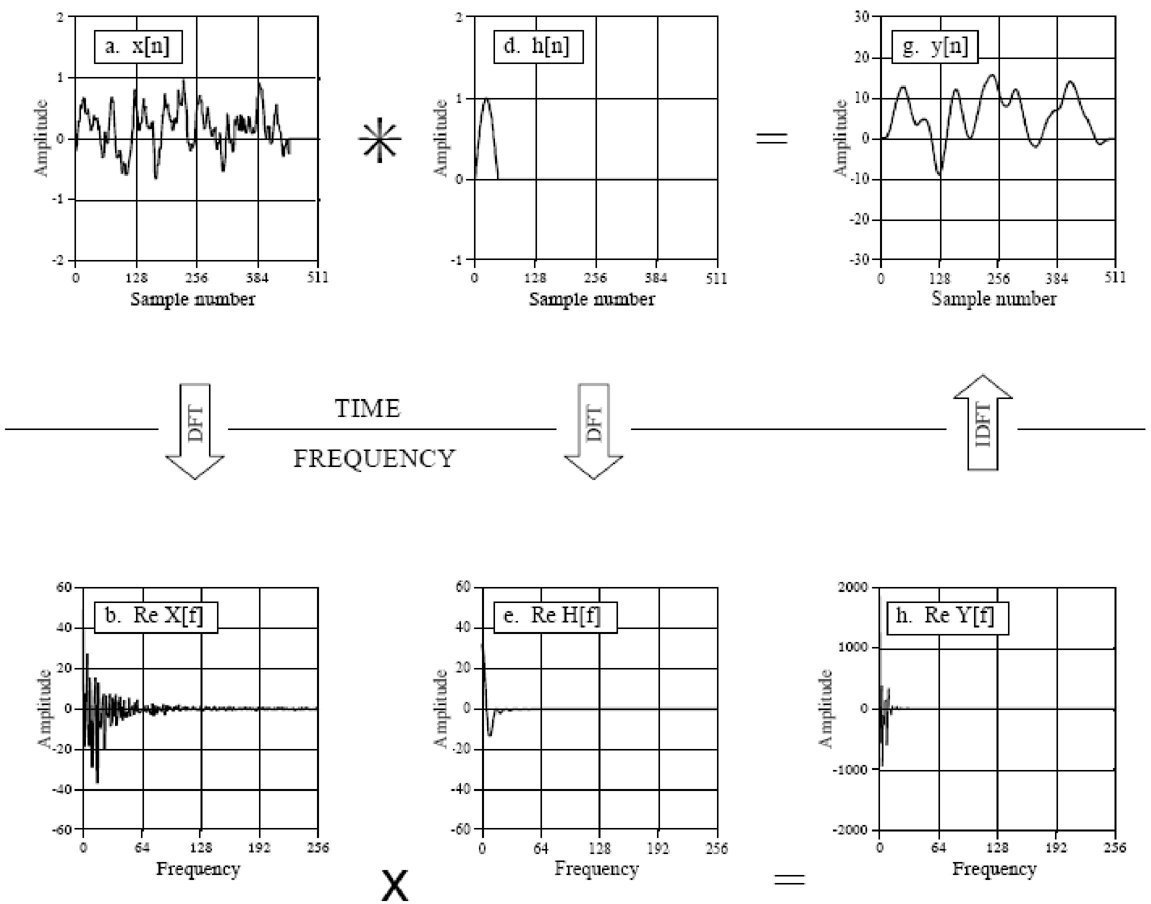




Time Domain/Freq Domain Duality

- Convolution in the time domain is equivalent to multiplication in the frequency domain
- Multiplication in the time domain is equivalent to convolution in the frequency domain

FFT Convolution



Convolution Via The Frequency Domain

- Convolution of two signals can be performed in the frequency domain
  - Multiply point by point the frequency domain representations of the signal
  - Convert back to the time domain using inverse DFT
- Requires special care to avoid circular convolution which produces corrupted results.

Summary - 1

- The frequency resolution of the DFT is determined by the number of samples

$$f_{res} = \frac{f_s/2}{N/2}$$

- The frequency resolution can be improved by
  - Using more data points
  - Adding zeros to the data set – zero padding.

Summary - 2

- Increasing the number of samples in the DFT improves frequency resolution but <u>does not</u> lower the noise level.

- The noise can be decreased by averaging multiple DFTs
  - The noise decreases by $1/\sqrt{N}$

54

Summary - 3

- Windowing modifies the amplitude of data points near the beginning and end of a data set
  - It reduces spectral leakage due to the finite number of data points in the data set.

- Windowing decreases frequency resolution but the benefits often outweigh this cost.