Digital Signal Processing

Final Exam ICPs

ICP 1 -- SNR

- In the signals below, the signal of interest is the mean or standard deviation of each signal.
- The noise level is also given as a standard deviation.
- Compute the signal to noise ratio for each signal in decibels.
    - $\mu_{signal} = 0.75V,\ \sigma_{noise} = .65\text{V}$
    - $\sigma_{signal} = 1.5V, \sigma_{noise} = .18V$

NOPRINT

SNR

- Signal 1 -- $\mu_{signal} = 0.75V,\ \sigma_{noise} =$ .65V

$$SNR = 20\log_{10}\left(\frac{\mu_{signal}}{\sigma_{noise}}\right) = 20\log_{10}\frac{.75V}{.65V} = 1.243\ dB$$

- Signal 2 -- $\sigma_{signal} = 1.5V, \sigma_{noise} = \ .18V$
-

$$SNR = 20\log_{10}\left(\frac{\sigma_{signal}}{\sigma_{noise}}\right) = 20\log_{10}\frac{1.5V}{.18V} = 18.42\ dB$$

ICP-2 Oversampling Averaging

- The noise level of a sensor as measured by its standard deviation is 15 mV. The sensor is being read by an ADC with a full-scale range of 5V and 8 bits of resolutions. Dither noise with a standard deviation of 15 mV is being added to the input
- I only need to sample the input at a rate of 1kHz, however the ADC system can be sampled as fast as 1 MHz
- If I use oversampling and averaging how fast should I sample the signal to make sure that the total noise of my signal is less than an equivalent value of 1 mV.

NOPRINT

Oversampling Averaging

- Compute the total noise input from the sensor, quantization and dither noise.

$$\sigma_q = 0.29 \times 1\ CV \qquad 1\ CV = \frac{V_{fs}}{2^N - 1} = \frac{5\mathrm{V}}{2^8 - 1} = 19.61\ \mathrm{mV}$$

ADC Noise

$$\sigma_q = 0.29 \times 19.61\ mV = 5.69mV$$

Dither noise → $\sigma_d = 15mV$

$$\sigma_{total} = \sqrt{\sigma_{signal}^2 + \sigma_d^2 + \sigma_q^2} = \sqrt{15^2mV + 15^2mV + 5.69^2mV}$$

Total Noise

$$\sigma_{total} = 21.96\ mV$$

NOPRINT

Oversampling Averaging

- Oversampling and averaging will reduce the standard deviation of my sample average by

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

- I want the σ of the sample averages to be less than 1 mV

$$N = \left(\frac{\sigma_{total}}{\sigma_{ave}}\right)^2 = \left(\frac{21.96\ mV}{1\ mV}\right)^2 = 482\ samples$$

I would need to sample at least 482 kHz and average those samples to achieve the noise requirements

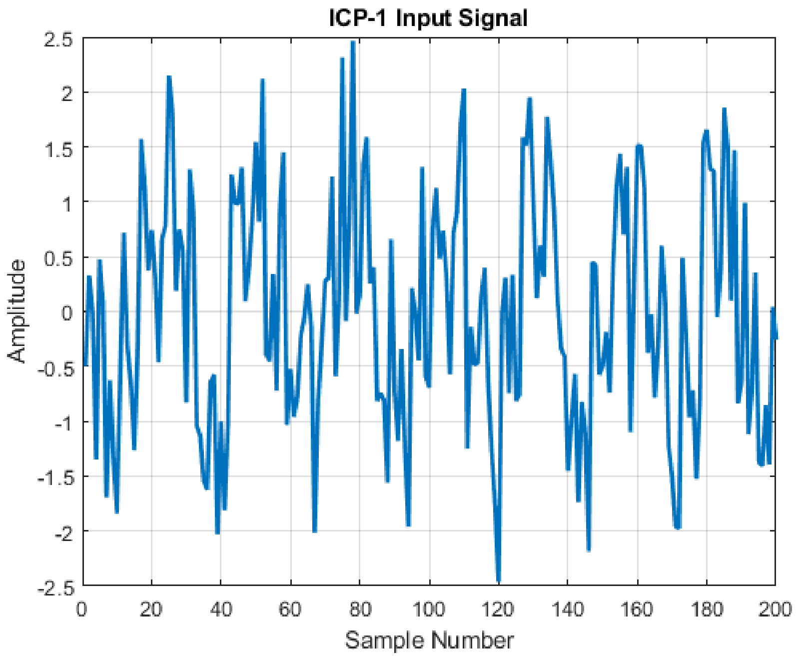
ICP-3
Frequency Domain Convolution

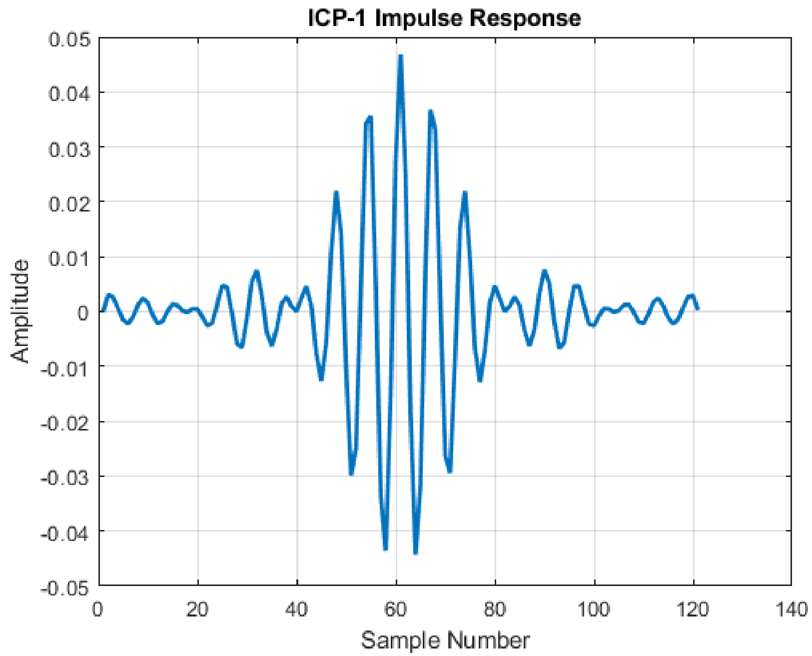
- Perform Frequency Domain Convolution between the two signals in the file ‘ICP_1.mat’ from my Courses

- Plot the Signal
- Plot the Impulse Response
- Pad the sequences to the correct length and take the FFT
- Multiply point by point
- Take the IFFT and truncate to the correct length

Time Domain
Signal and Impulse Response

NOPRINT





200 samples

121 samples

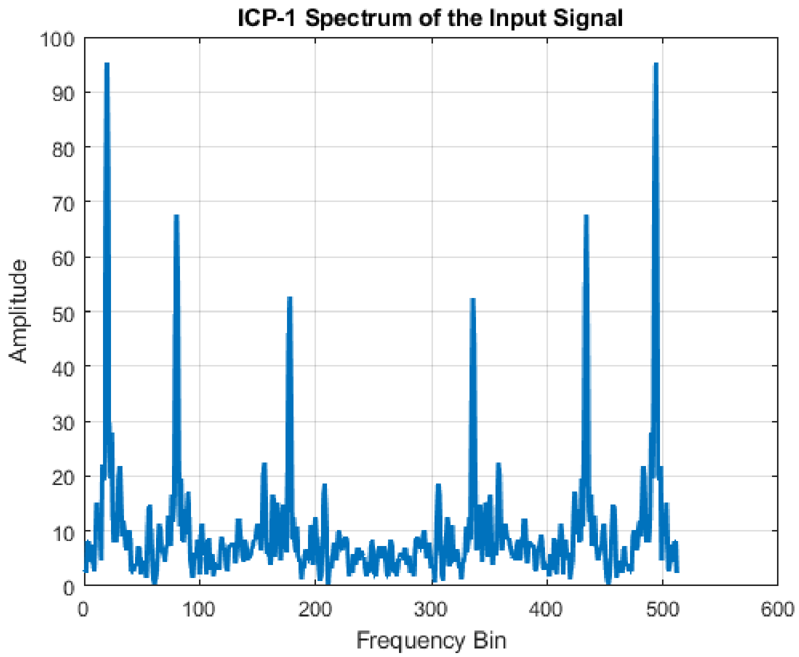
Convolution Length = 200 samples + 121 samples – 1 = 320

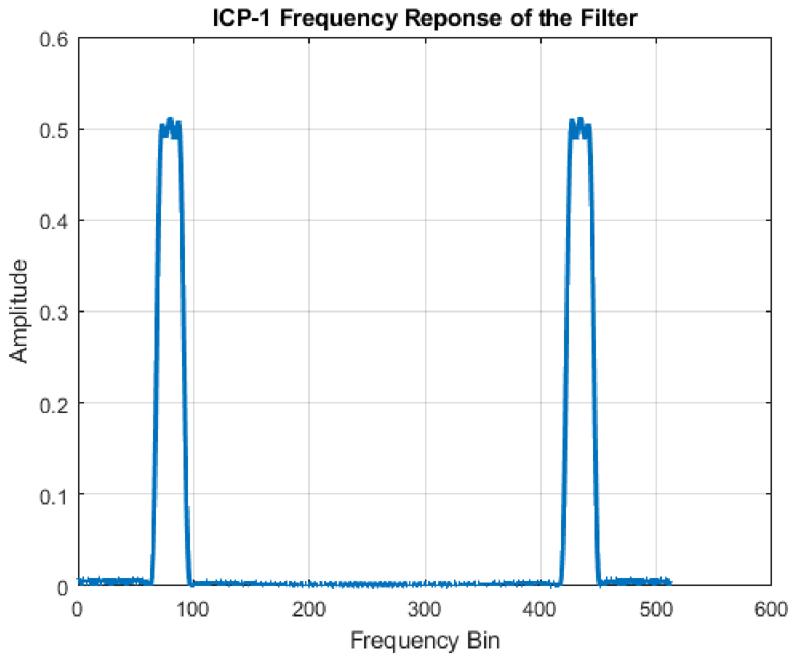
Frequency Domain
Spectra and Frequency Response

NOPRINT

Pad to the next highest power of 2 -- 512 samples

Take the FFT of the signal and the impulse response

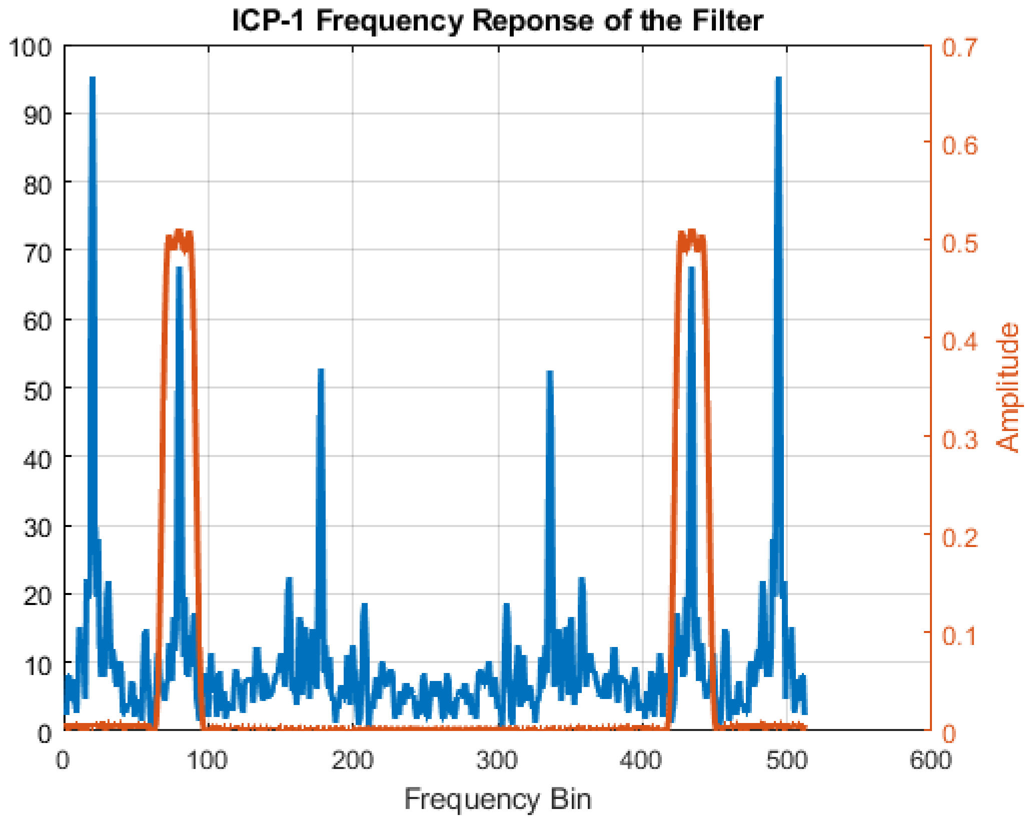




Plot the magnitudes, but keep the complex values

Frequency Domain Spectra and Frequency Response

NOPRINT



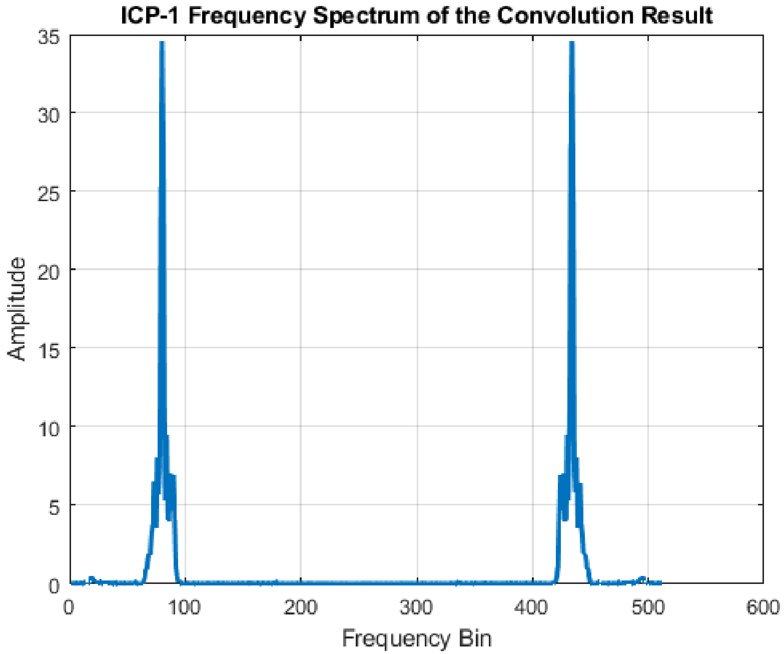
Multiply the complex values point by point (use .*)

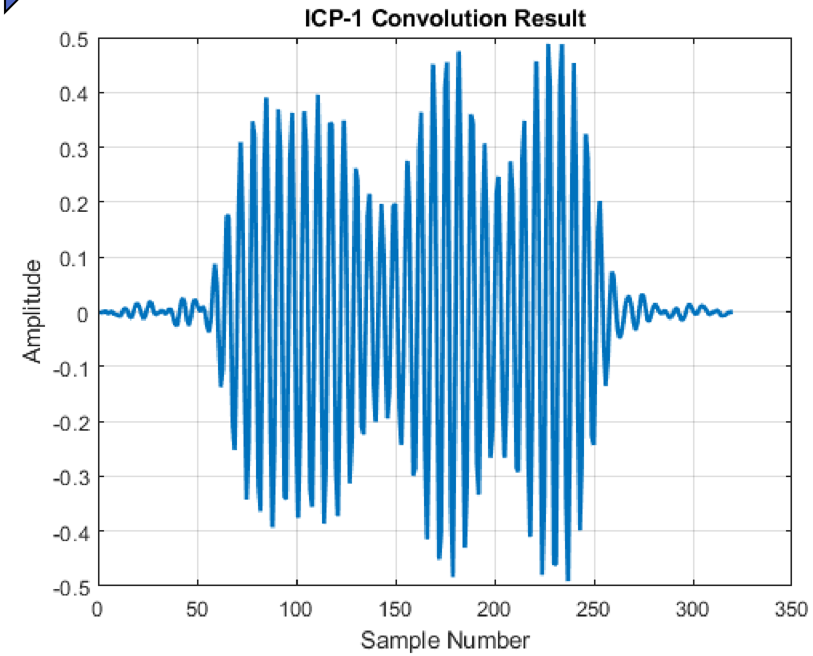
Convolution Output
Frequency and Time Domain

NOPRINT

Result of the multiplication

Take the IFFT, keep 320 samples





ICP-4
Windowed SINC Filter

- Using FIR Designer design a Windowed SINC LPF with a corner frequency of 30 BPM with a transition region that is 10 BPM wide

Calling FIR Designer

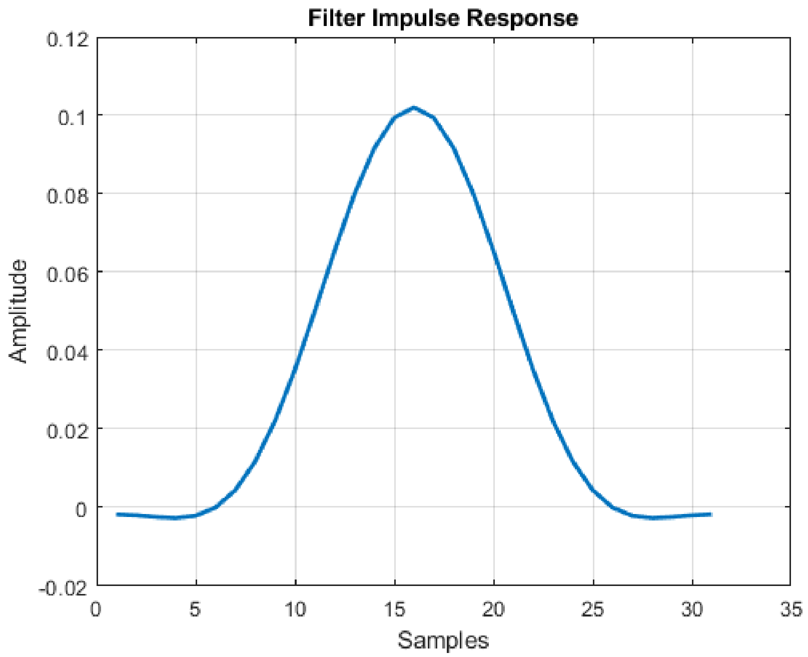
NOPRINT

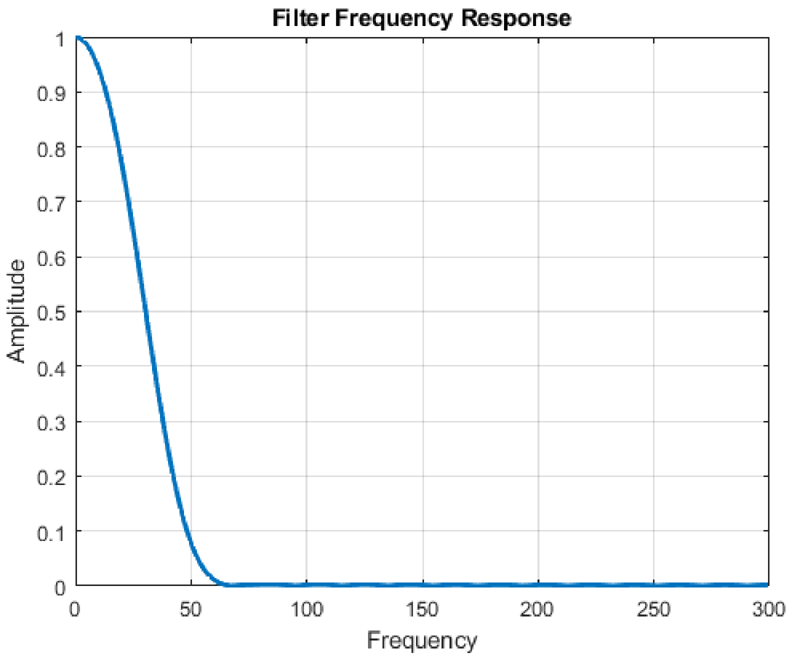
```
fcLPF = 30;
lpfLength = 31;   %  Start with 31 and check the response

%  User FIR_Designer.  Turn off the header printout

if answersOn

hLPF = FIR_Designer('nOrder',lpfLength, 'cutBPM', fcLPF, 'PrintHeader',false);
fcLPF = 30;
lpfLength = 111;   %  Iterated through a few options and settled on this.

%  User FIR_Designer.  Turn off the header printout

hLPF = FIR_Designer('nOrder',lpfLength, 'cutBPM', fcLPF, 'PrintHeader',false);
end
```

Iterate through different lengths to find the proper length

Length 31 FIR

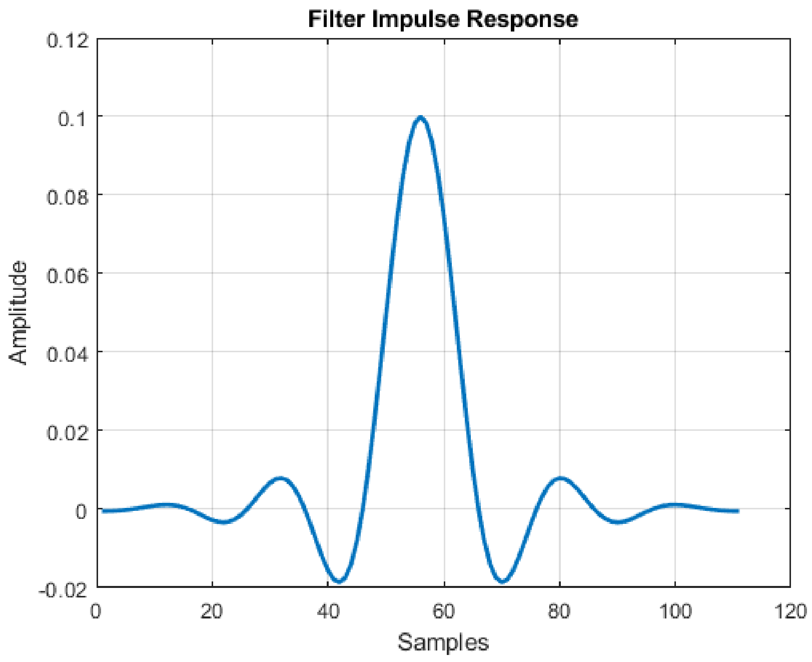
NOPRINT

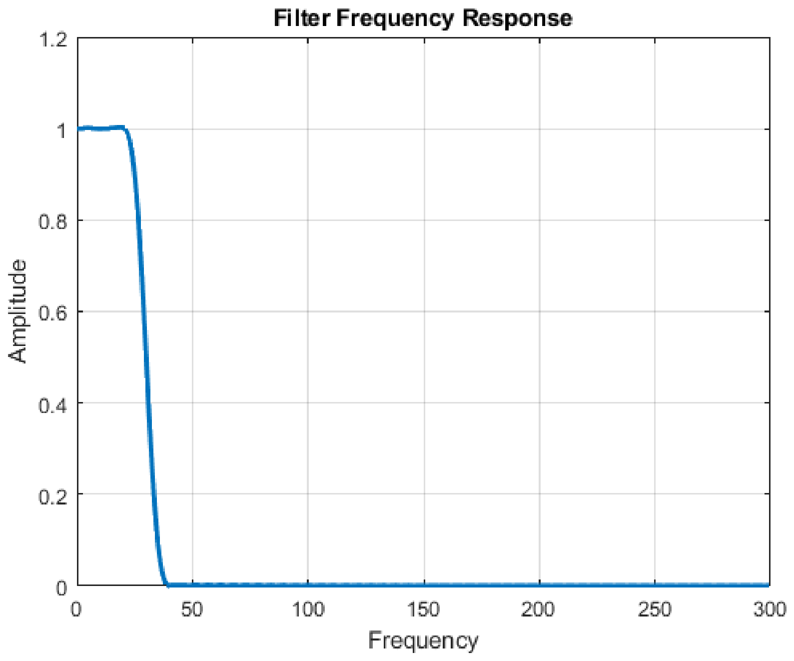




Length 111 FIR

NOPRINT





ICP 5 - Moving Average Filter

- What is the impulse response of a moving average filter of kernel length 21

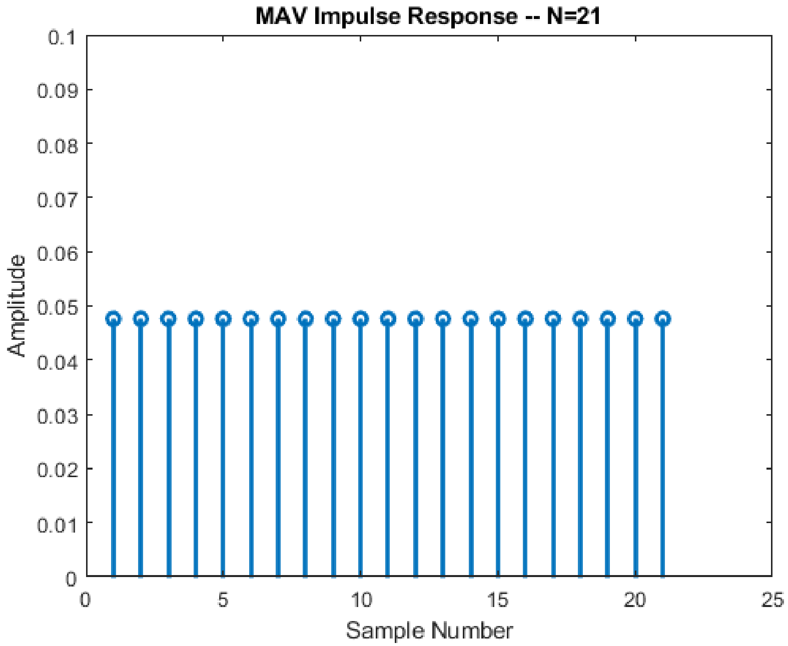
- What is the frequency response of the filter?

Moving Average Filter

NOPRINT

- The moving average filter has an impulse response of all the same values N for example

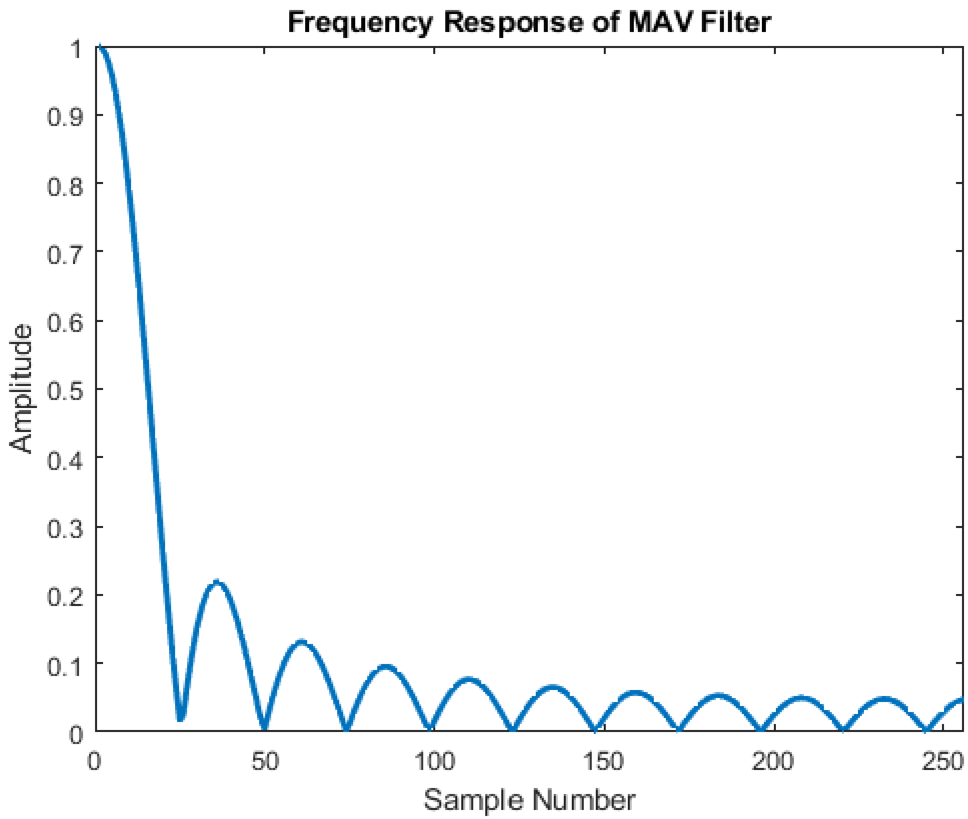
- To have a DC gain of zero each values is 1/N



Moving Average Filter

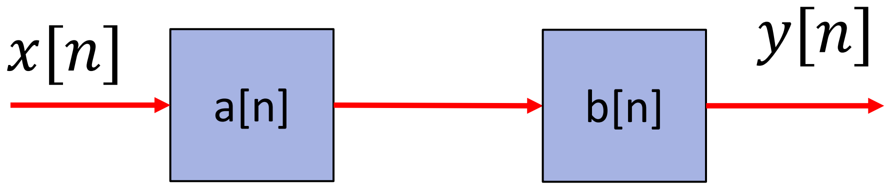
NOPRINT

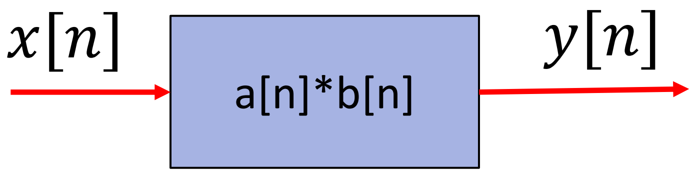
- The frequency response is a SINC function



ICP – 6 Properties of Convolution

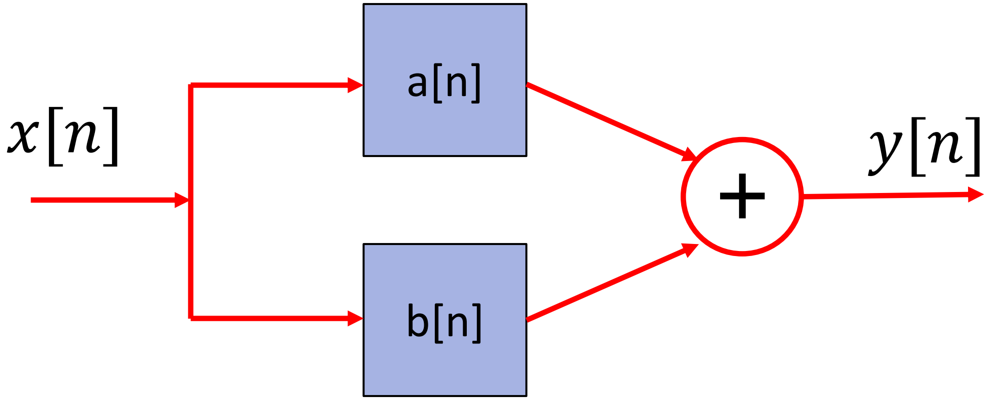
- There are two impulse responses
  - $a[n] = [0.25,\ .5,\ 0.25]$ and $b[n] = [\,-1, 1,\ 2\,]$
- Find their cascaded response
- Find the output of the cascaded system when
  - $x[n] =$ [1, 0, 1, 0, 1]

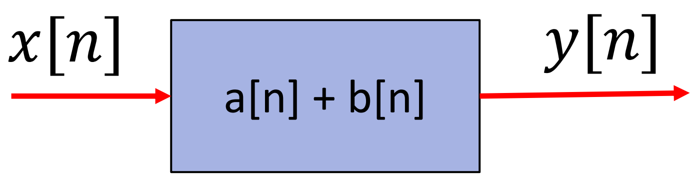




Properties of Convolution
Distributive Property

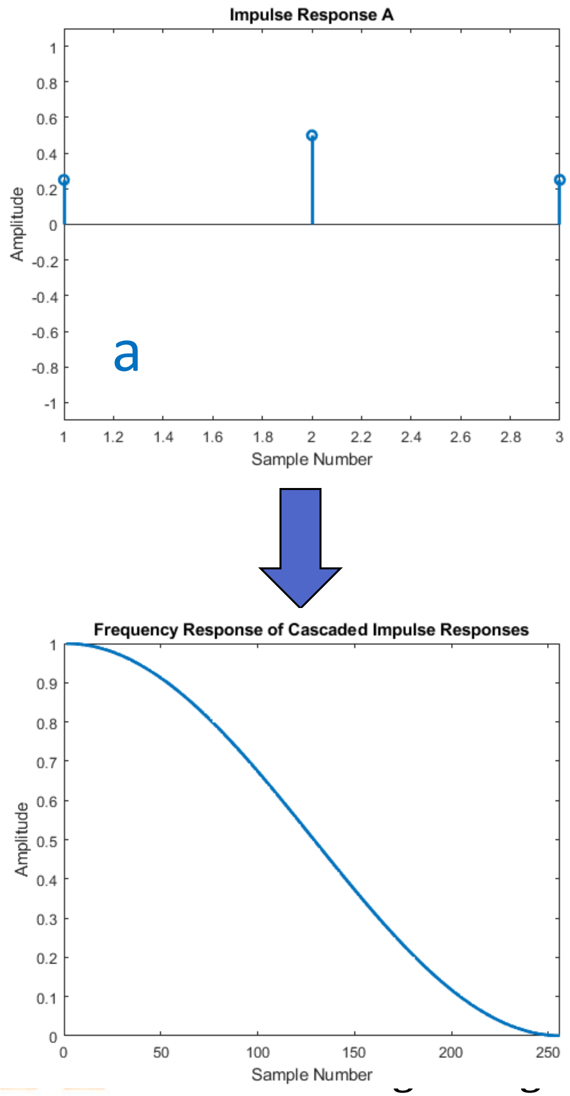
$$x[n] * a[n] + x[n] * b[n] = x[n] * \left[a[n] + b[n]\right]$$

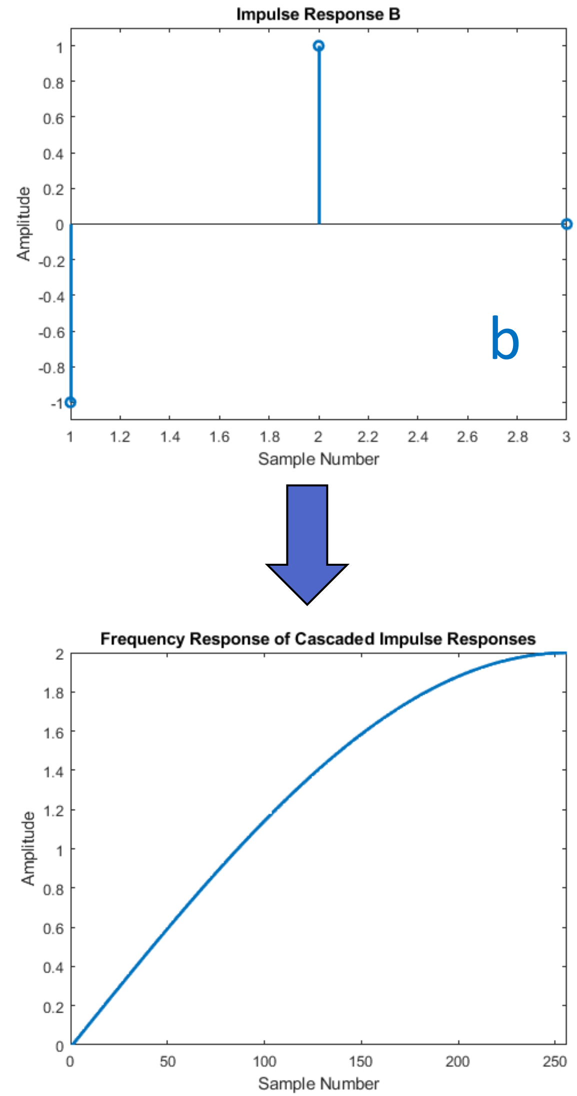




Individual Responses Cascaded

NOPRINT

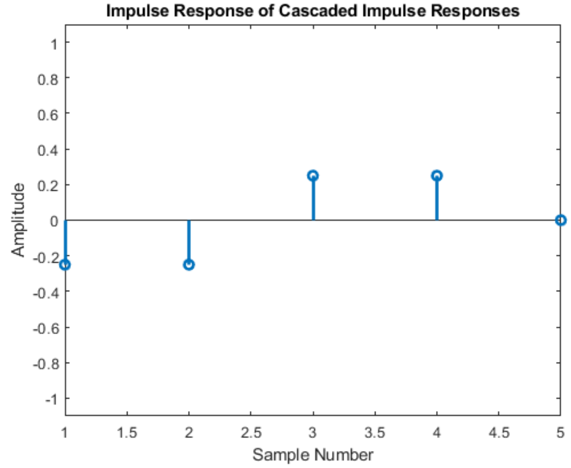


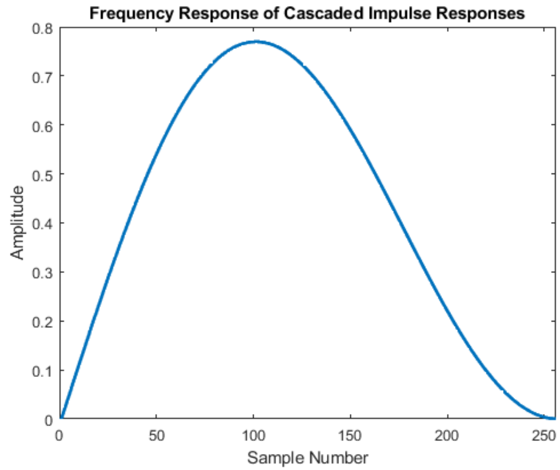


Individual Responses Cascaded

NOPRINT

h = a*b

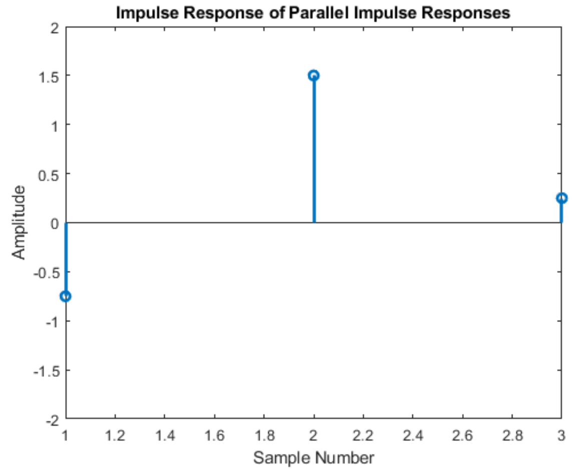


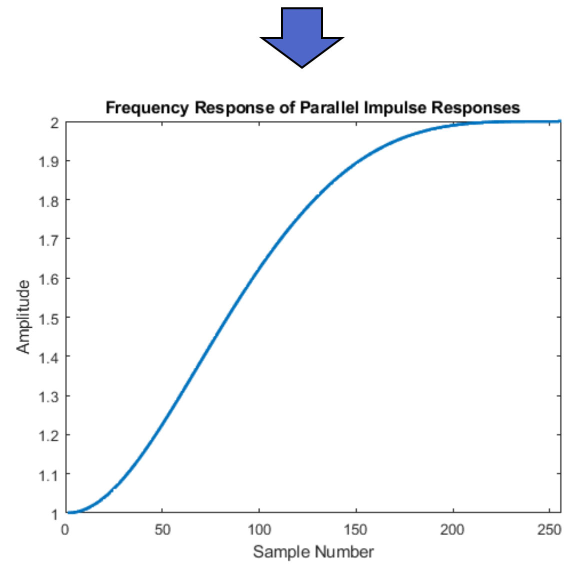


Individual Responses In Parallel

NOPRINT

h = a+b





ICP-7
Single Pole IIR LPF and HPF

- Design a single pole IIR filter with a corner frequency of 250 Hz. The sampling rate of the system is 8000 Hz. Plot the impulse response and the frequency response of the filter.

Low Pass Filter

NOPRINT

Calculate the normalized corner frequency

$$f_{norm} = f_c / f_s \qquad f_{norm} = 0.03$$

Find the value of x, the decay from sample to sample

$$x = e^{-2\pi f_{norm}} \qquad x = 0.82$$

Compute the value of b1 and a0

$$b_1 = x \qquad b_1 = 0.82$$

$$a_0 = 1 - x \qquad a_0 = 0.18$$

Find the Impulse Response

NOPRINT

- Use "impz" function in MATLAB

$$H(z) = \frac{a_0}{1 - b_1 z^{-1}}$$

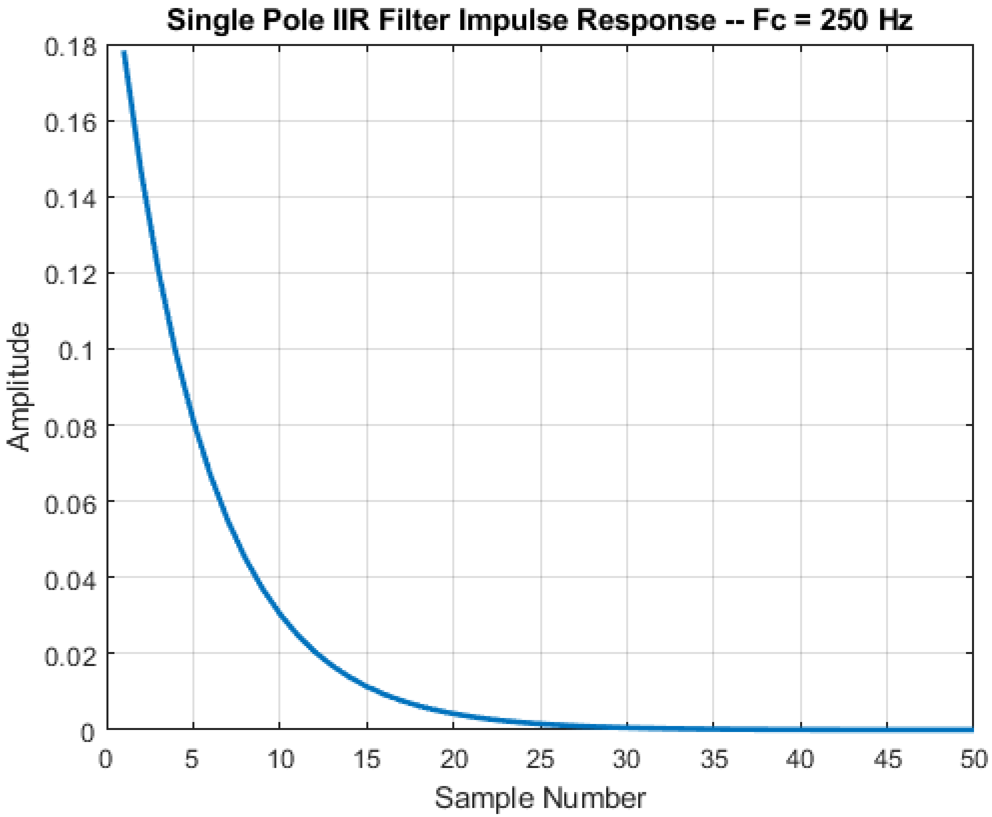
Numerator = $[a_0]$

Denominator = $[1, -b_1]$

```
num = a0;
den = [1, -b1];
impulseResponse = impz(num, den, 50);
```

Impulse Response

NOPRINT



Find the Frequency Response

NOPRINT

- Use "freqz" function in MATLAB

$$H(z) = \frac{a_0}{1 - b_1 z^{-1}}$$

Numerator = $[a_0]$

Denominator = $[1, -b_1]$

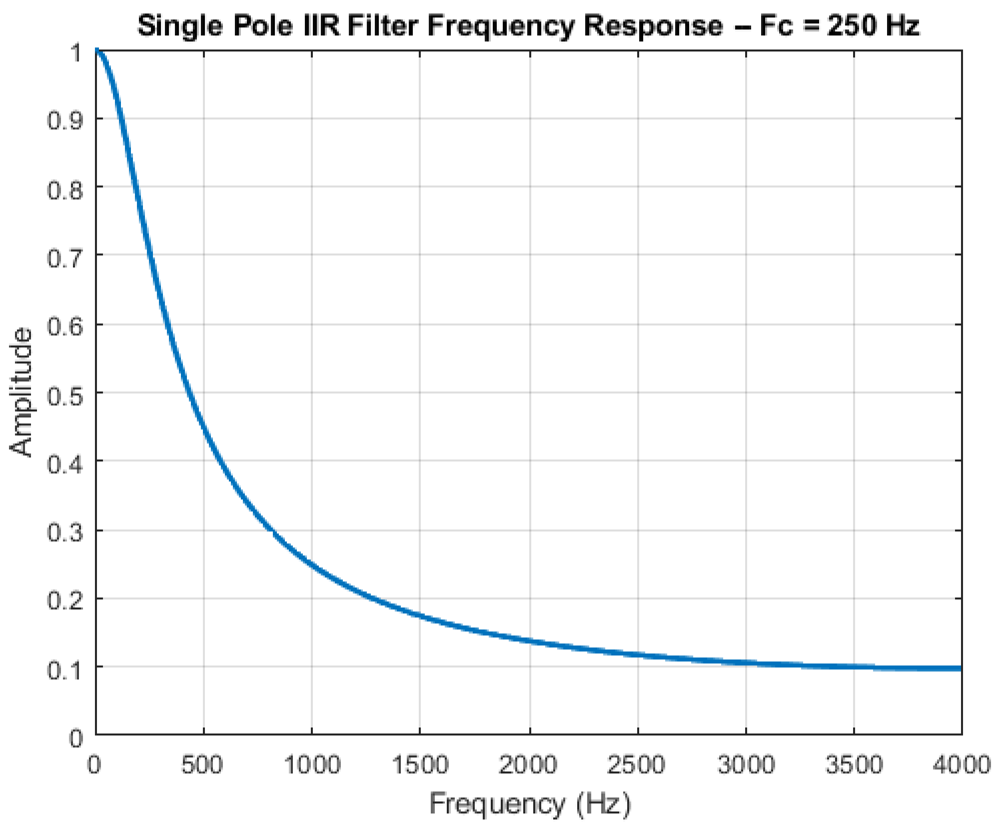
```
[frequencyResponseMag, freqReponseFreq] = freqz( num, den, 512 );
```

This function computes the frequency response from 0 to Nyquist $(f_s/2)$

Number of points in the frequency response

Frequency Response

NOPRINT



Single Pole IIR HPF

NOPRINT

- Similar approach to the LPF, just different coefficients

Calculate the normalized corner frequency

$$f_{norm} = f_c / f_s \qquad f_{norm} = 0.13$$

Find the value of x, the decay from sample to sample

$$x = e^{-2\pi f_{norm}} \qquad x = 0.46$$

Compute the value of b1 and a0

$$b_1 = 0.46$$

$$a_0 = (1 + x)/2 \qquad a_0 = 0.73$$

$$b_1 = x$$

$$a_1 = -(1 + x)/2 \qquad a_1 = -0.73$$

Find the Impulse Response of the HPF

NOPRINT

- Use "impz" function in MATLAB

$$H(z) = \frac{a_0 + a_1 z^{-1}}{1 - b_1 z^{-1}}$$

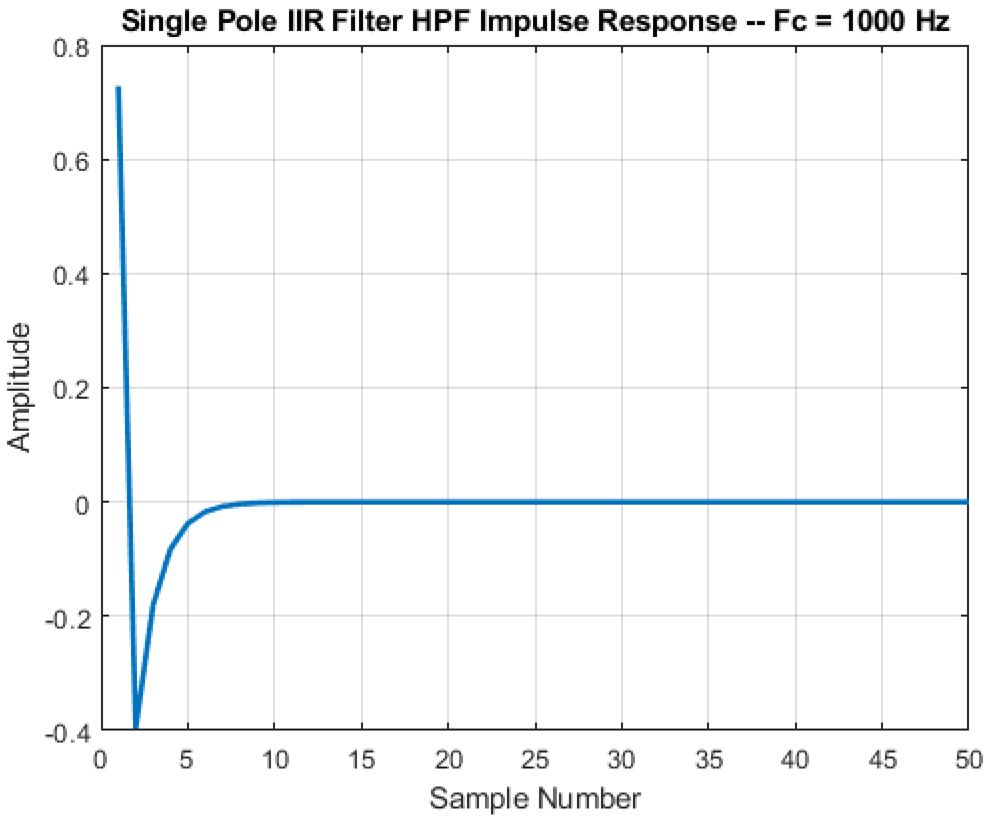
Numerator = $[a_0, a_1]$

Denominator = $[1, -b_1]$

```
num = [a0, a1];
den = [1, -b1];
impulseResponseHPF = impz(num, den, 50);
```

HPF Impulse Response

NOPRINT



Find the Frequency Response of the HPF

NOPRINT

- Use "freqz" function in MATLAB

$$H(z) = \frac{a_0 + a_1 z^{-1}}{1 - b_1 z^{-1}}$$

Numerator = $[a_0, a_1]$

Denominator = $[1, -b_1]$

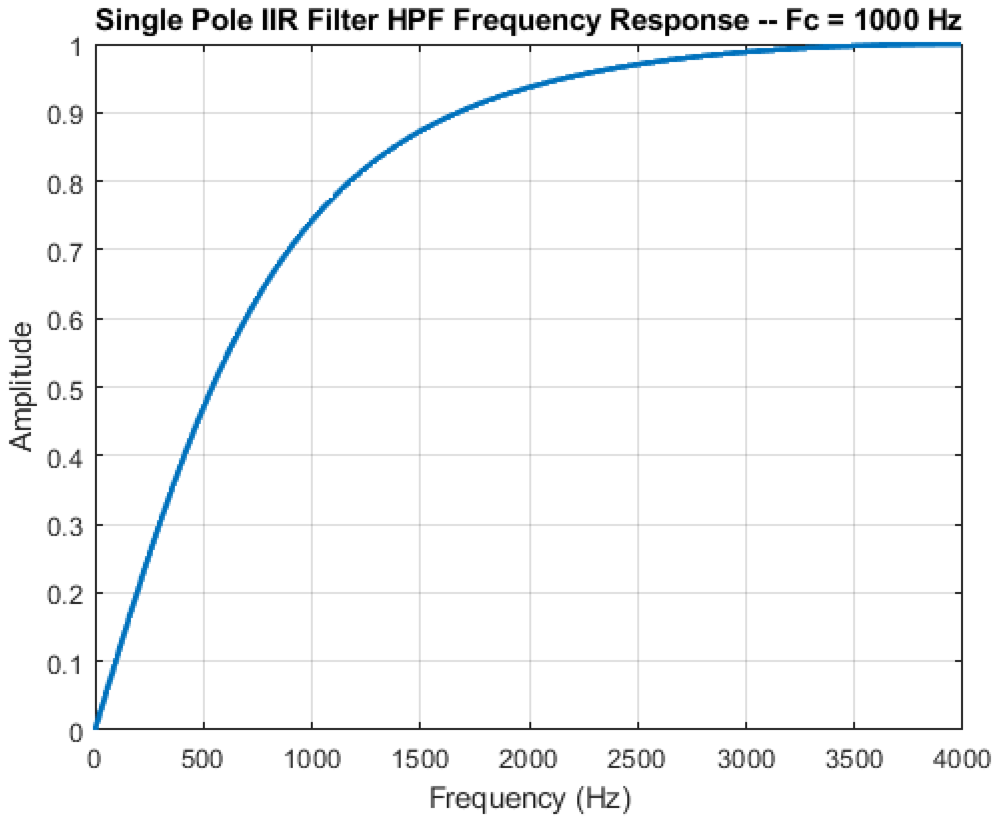
```
%  Find the frequency response using freqz

[frequencyResponseMag, freqReponseFreq] = freqz( num, den, 512 );
```

This function computes the frequency response from 0 to Nyquist $(f_s/2)$

Number of points in the frequency response

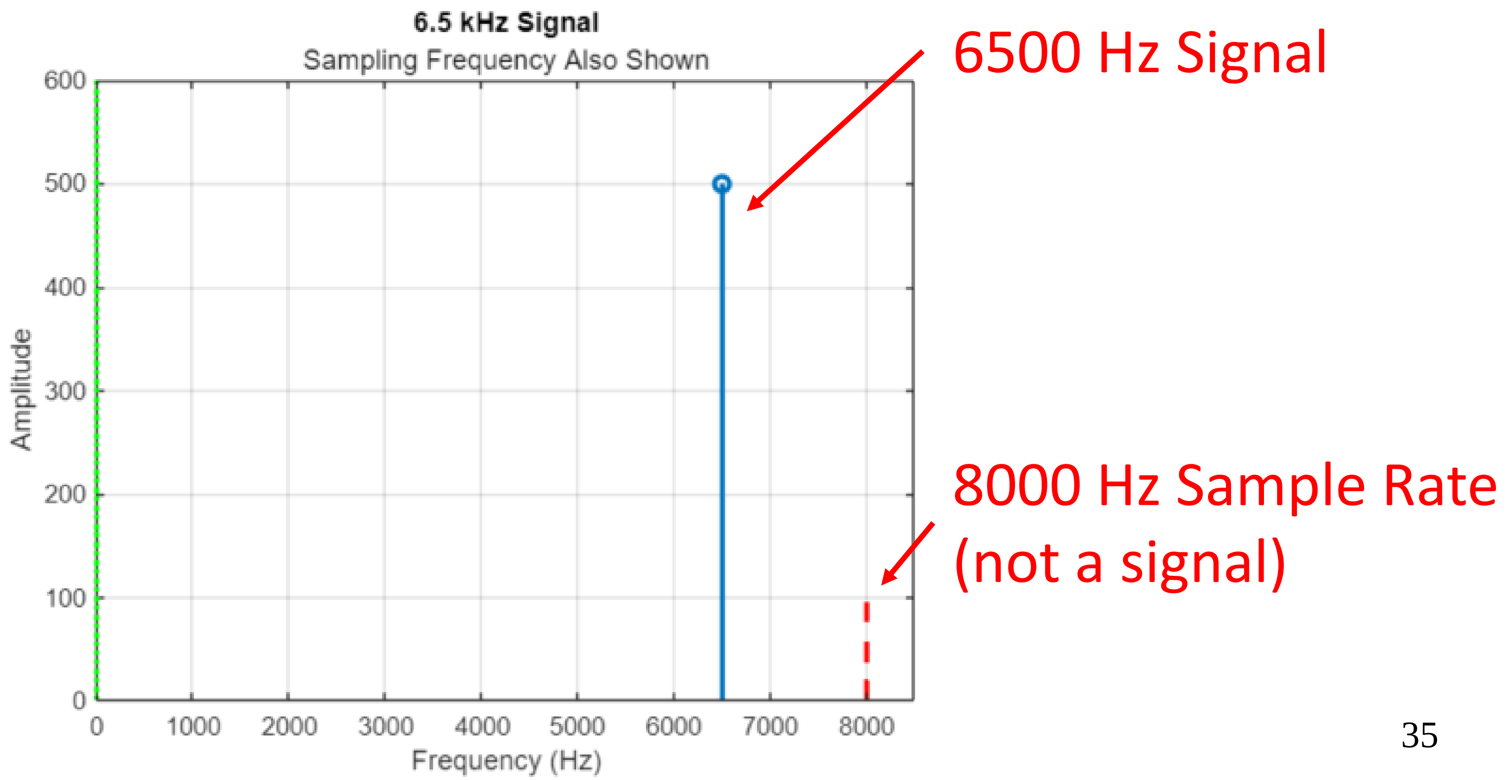
Frequency Response of the HPF

NOPRINT



ICP-8
Sampling and Aliasing

- A sine wave of 6500 kHz is being sampled at a sampling rate of 8 kHz. After sampling at what positive frequencies will the first 4 copies of the sinusoids be located?

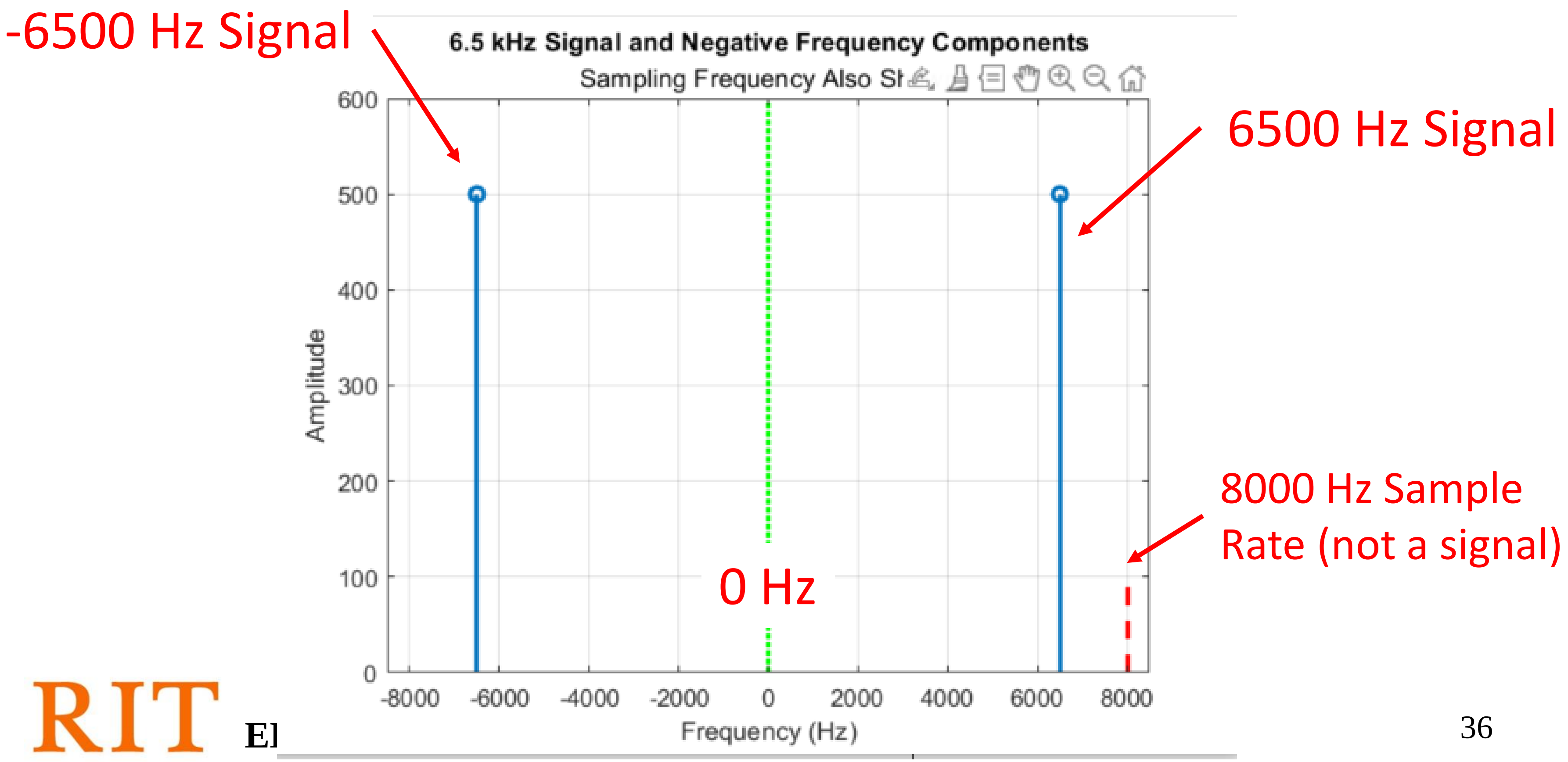


RIT

Sampling and Aliasing

NOPRINT

- Recall that for real signals, for each positive frequency there is a corresponding negative frequency



RIT El

Sampling and Aliasing

NOPRINT

- Sampling causes copies of the signal to be created at intervals of the sampling rate. The positive frequencies that the 6.5 kHz signal will be copied to are

6.5 kHz + 8 kHz = 14.5 kHz

14.5 kHz + 8 kHz = 22.5 kHz

6.5 kHz - 8 kHz = -1.5 kHz

-1.5 kHz - 8 kHz = -9.5 kHz

Sampling and Aliasing

NOPRINT

- There is a negative frequency component at -6.5 kHz and this will be repeated every 8 kHz as well

-6.5 kHz + 8 kHz = 1.5 kHz

1.5 kHz + 8 kHz = 9.5 kHz

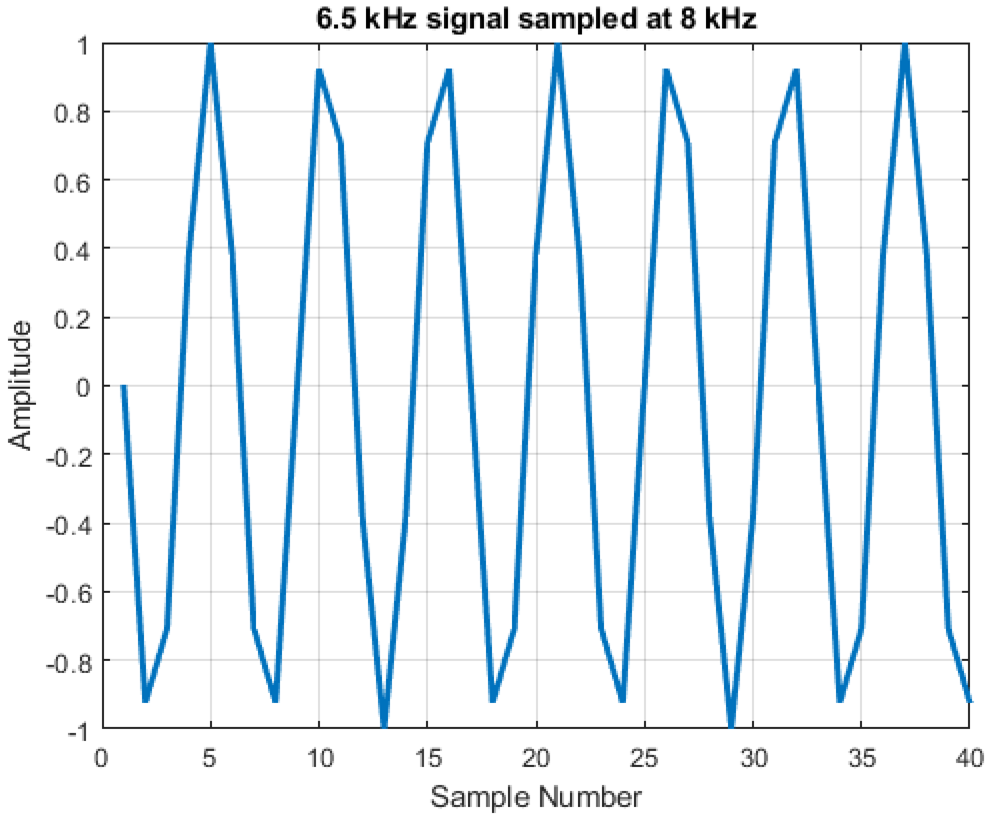
-6.5 khz - 8 kHz = -14.5 kHz

-14.5 kHz - 8 kHz = -22.5 kHz

Sampling and Aliasing

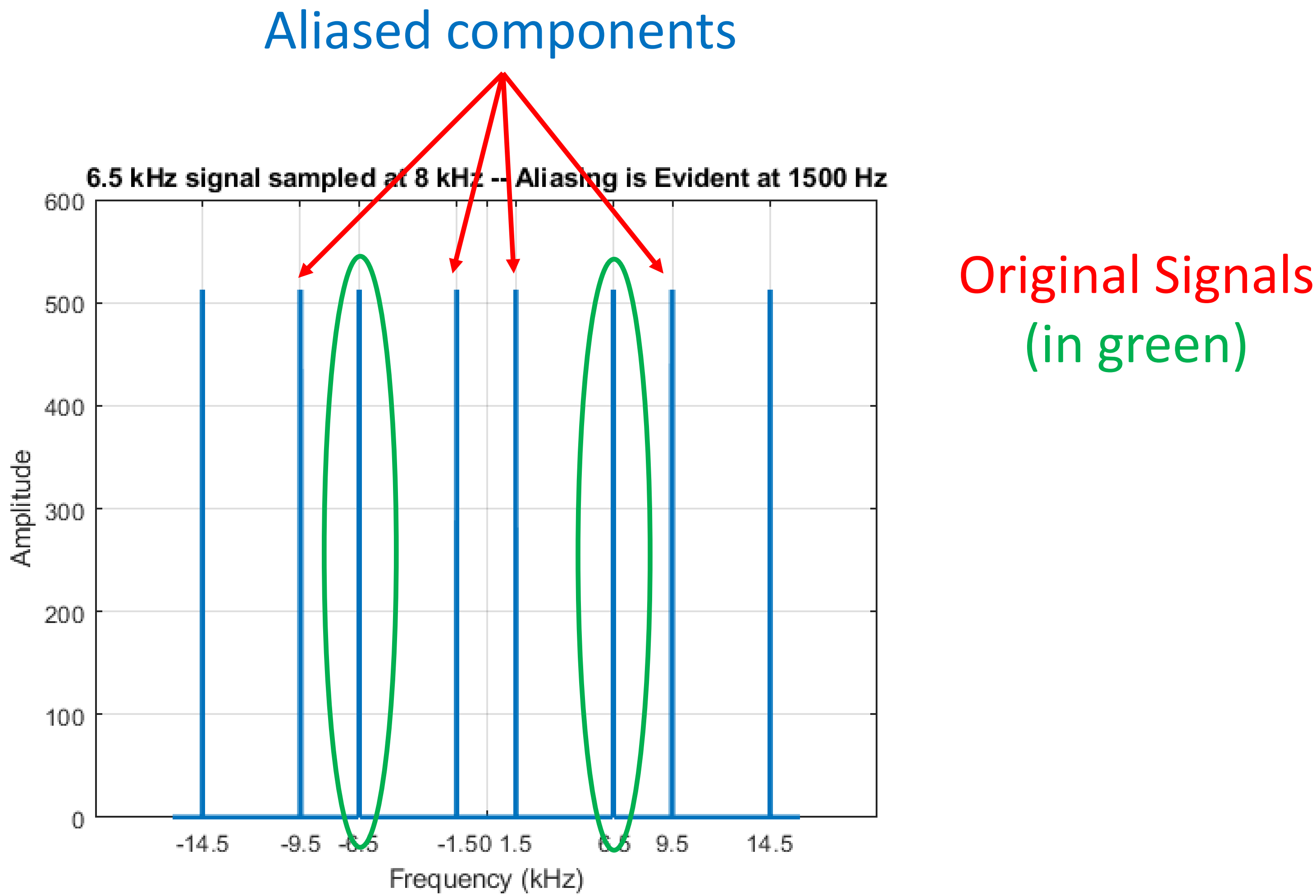
NOPRINT

- Sample a 6.5 kHz sine wave at 8 kHz



Spectrum Showing Aliasing

NOPRINT

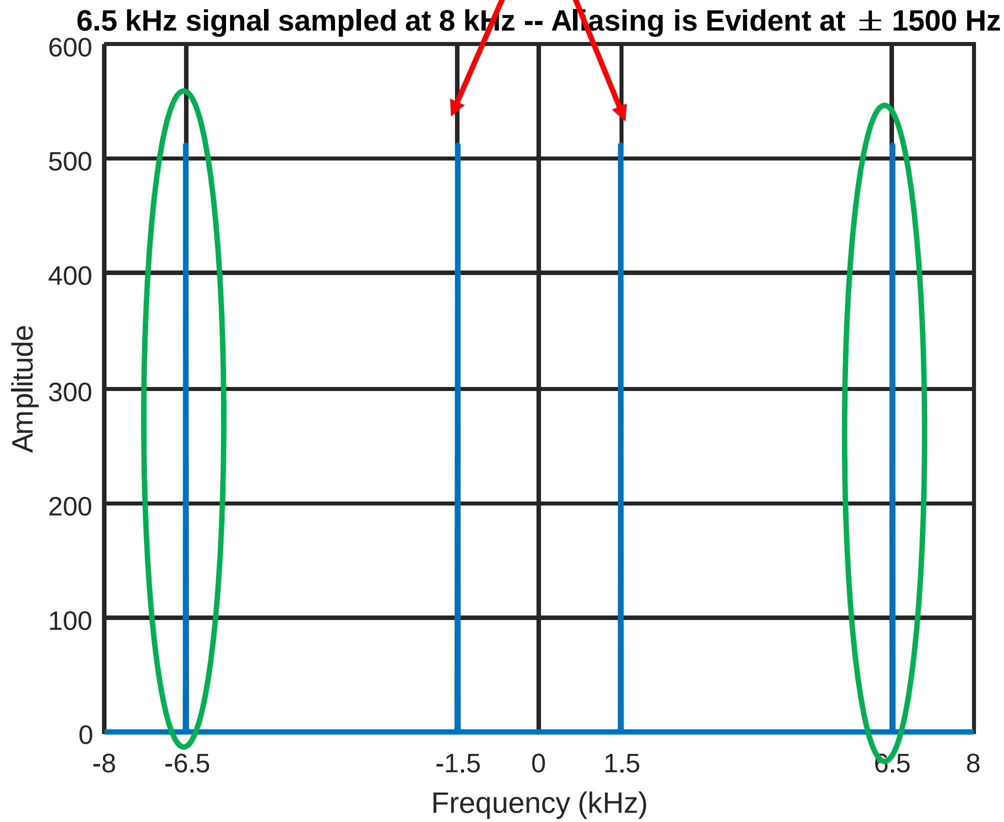


Spectrum Close Up (Plotted $\pm 8\ kHz$)

NOPRINT

Aliased components

Original Signals



ICP-9
Sampling and Quantization Noise

- I'm sampling a sinewave with an amplitude of 100 mVrms. The sinewave has a noise level of 2.9 mVrms on it.
- I plan on sampling the signal with an ADC that has an input range of -3V to +3V.
- What resolution ADC do I need to degrade the signal to noise ratio by no more than 8 dB?

ICP-9
Sampling and Quantization Noise

- Steps
  - Find the SNR of the input signal in dB
  - Compute the degraded SNR in dB (-8)
  - Find the degraded SNR numerically
  - Find the total amount of noise for the degraded SNR
  - Compute the maximum amount of quantization noise that could be added
  - Determine the number of bits in the ADC that results in that amount of quantization noise

Steps 1-3

NOPRINT

The input signal SNR is

$$SNR_{dB} = 20\log_{10}\frac{100\ mV_{rms}}{2.9\ \ mV_{rms}} = 30.75\ dB$$

The degraded SNR (in dB) is

$$SNR_{degdB} = SNR_{dB} - 8 = 22.75$$

The degraded SNR (numerically) is

$$SNR_{deg} = 10^{\frac{SNR_{\mathrm{deg}dB}}{20}} = 13.72$$

Steps 4-5

NOPRINT

The total noise of the degraded signal is

$$\sigma_{total\_noise} = \frac{\sigma_{signal}}{SNR_{degraded}}$$

Find the total quantization noise allowed

$$\sigma_{total} = \sqrt{\sigma_{input\_noise}^2 + \sigma_q^2}$$

Solve for σ_q

$$\sigma_q = \sqrt{\sigma_{total}^2 - \sigma_{input\_noise}^2} = \sqrt{7.28^2 mV - 2.9^2 mV} = 6.67\ mV$$

Step 6

NOPRINT

Find the number of bits in the ADC

Full scale of the ADC

$$\sigma_q = 0.29 \times \frac{6V}{2^N - 1}$$

Number of levels

Solve for N

$$2^N = 0.29 \times \frac{6V}{6.67mV} + 1$$

$$N = \log_2\left(0.29 \times \frac{6V}{6.67mV} + 1\right) = 8.03\ bits$$

Move up to 9 bits

ICP-10
Higher Order Recursive Filters

- Design a 5 pole IIR Chebyshev lowpass filter with 0.5 dB of ripple that has a corner frequency of 70 BPM using a sample rate of 600 BPM.
- Compute the filter coefficients using the MATLAB IIR_Designer tool.
- Plot the impulse and the frequency response in MATLAB. Use the MATLAB scripts provided to you to design IIR filters. Use the direct form coefficients from MATLAB.

ICP-10
Higher Order Recursive Filters

Typically the breathing rate system is using a sampling rate of 10

Set the sampling frequency (e.g. 600)

samplingFreq 600

Choose the filter type (Butterworth or Chebyshev) and the filter

Set the filter type (e.g. Chebyshev)

filterType Chebyshev

filterTopology Lowpass

Set the filter topology (e.g. Lowpass)

An additional parametr for the Chebyshev filter is the amount of ripple in the passband. If Chebyshev i

rippleDb 0.5

Set the filter ripple (Chebyshev only)

The order of the filter will determine the number of poles in th

filterOrder 5

Set the filter Order (e.g. 5)

Choose the corner frequency (sometimes called the cutoff frequencv). This determines the location of frequency and the upper corner frequency. The figures below

If the filter is a LPF or HPF, enter the corner freq

Set the filter topology (e.g 70) Lowpass and highpass

cornerFreq 70

If the filter is a BPF or BSF, enter both the upper

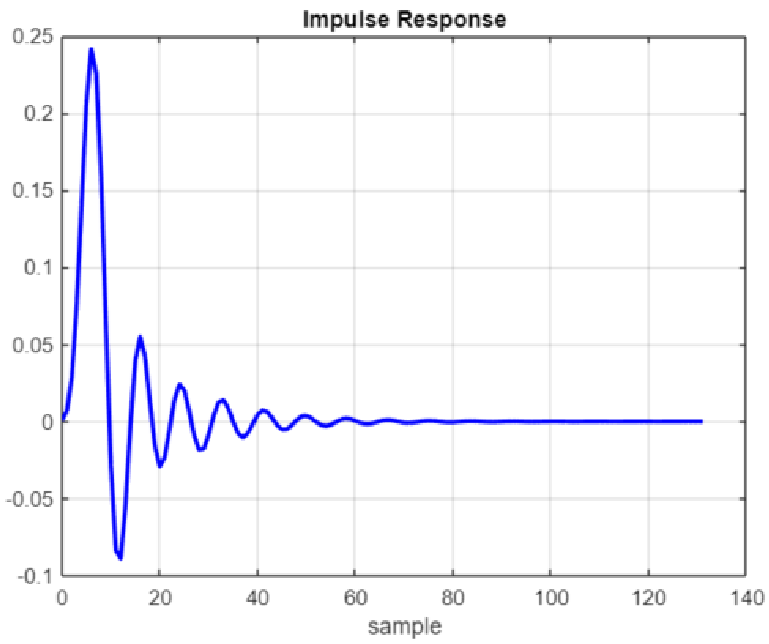
Corner frequencies for BPF and BSF

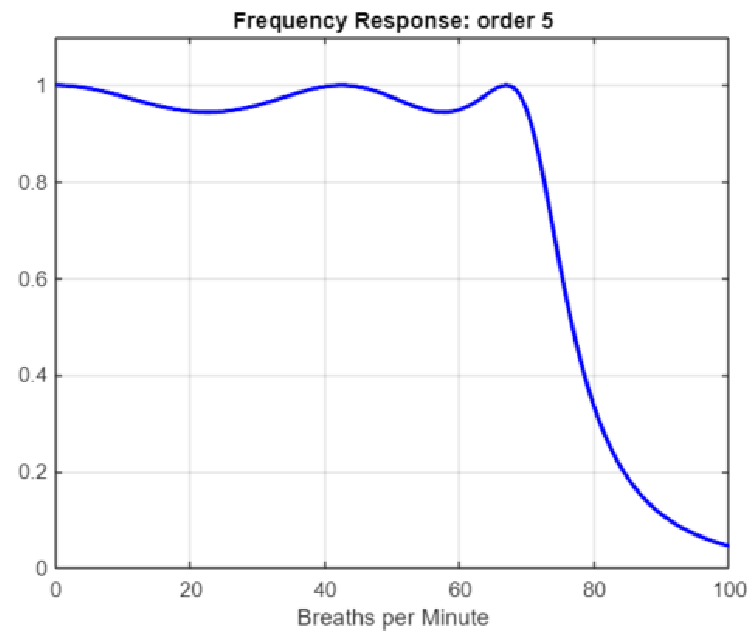
lower_bpf_bsf_cornerFreq 23

upper_bpf_bsf_cornerFreq 55

ICP-10
Higher Order Recursive Filters

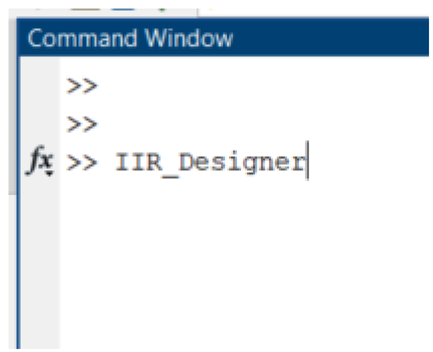
- The application will plot the impulse response and the frequency response of the filter





ICP-10
Higher Order Recursive Filters

- To use the filter coefficients in MATLAB
  - Set all the parameters in the script
  - IN THE COMMAND WINDOW execute the IIR_Designer command

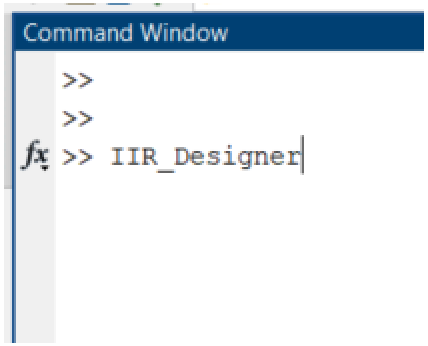


- To use the filter coefficients in MATLAB
  - Set all the parameters in the script
  - IN TI comm

```
Direct IIR Filter Coefficients for easy copy to MATLAB

b = [0.0008160, 0.0040802, 0.0081603, 0.0081603, 0.0040802, 0.0008160];
a = [1.0000000, -3.5596819, 5.5972100, -4.7578789, 2.1711919, -0.4247281];
```

ICP-10
Higher Order Recursive Filters

- Copy the coefficients from the command window and put them in your MATLAB code
  - Numerator coefficients is the “b” variable
  - Denominator coefficients is the “a” variable



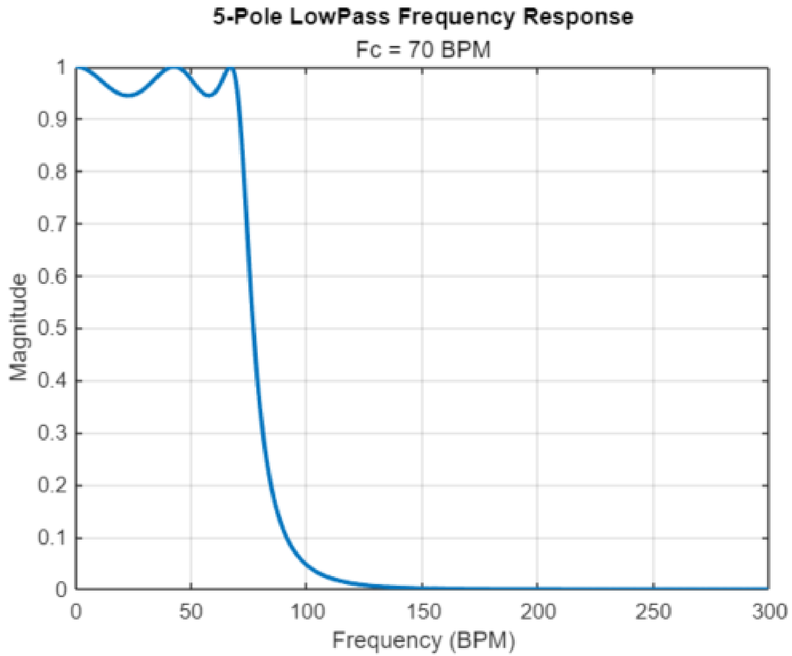
Numerator coefficients (b)

```
Direct IIR Filter Coefficients for easy copy to MATLAB

b = [0.0008160, 0.0040802, 0.0081603, 0.0081603, 0.0040802, 0.0008160];
a = [1.0000000, -3.5596819, 5.5972100, -4.7578789, 2.1711919, -0.4247281];
```

Denominator coefficients (a)

ICP-10
Higher Order Recursive Filters

```
b = [0.0008160, 0.0040802, 0.0081603, 0.0081603, 0.0040802, 0.0008160];
a = [1.0000000, -3.5596819, 5.5972100, -4.7578789, 2.1711919, -0.4247281];

[mag, freq] = freqz(b, a, 512);
sampleFreq = 600;

figure
plot( freq/(2*pi)*sampleFreq, abs(mag),'LineWidth',2);
grid on

title('5-Pole LowPass Frequency Response','Fc = 70 BPM')
xlabel('Frequency (BPM)')
ylabel('Magnitude')
```



- Frequency response computed using ‘freqz’ function