University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering

ECE 311: Digital Signal Processing Lab

LAB 5: **Frequency Response and Analysis of Discrete-Time Systems**
Fall 2015

---

1 Objective

The objective of this lab is to help you understand the frequency response of a discrete-time linear shift-invariant (LSI) system.

Key Concepts:

- Magnitude and Phase of the frequency response of an LSI system.
- Linear phase
- Finite (FIR) vs. Infinite (IIR) Impulse Response filters
- Learn to use the filter design tool, `fdesign`, in Matlab

2 Background

2.1 Frequency Response of an LSI System

An LSI system with input $x[n]$ and output $y[n]$ must satisfy

$$y[n] = \sum_{m=-\infty}^{\infty} h[m]x[n-m] \tag{1}$$

for all n and every input $x[n]$, where $h[n]$ is the impulse response of the LSI system. By taking the DTFT of both sides of the convolution sum, we obtain

$$\begin{aligned} Y_d(\omega) &= \sum_{n=-\infty}^{\infty} \left(\sum_{m=-\infty}^{\infty} h[m]x[n-m] \right) e^{-j\omega n} \text{ substitute } n = k+m \\ &= \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h[m]x[k]e^{-j\omega(k+m)} \\ &= \left(\sum_{m=-\infty}^{\infty} h[m]e^{-j\omega m} \right) \left(\sum_{k=-\infty}^{\infty} x[k]e^{-j\omega k} \right) \\ &= H_d(\omega)X_d(\omega) \end{aligned} \tag{2}$$

where $H_d(\omega)$ is the DTFT of the impulse response and is referred to as the frequency response of the LSI system.

Now consider an input $x[n] = e^{j\omega_0 n}$ that is known as an "eigensequence". Using the convolution sum, the output $y[n]$ can be derived as

$$\begin{aligned} y[n] &= \sum_{m=-\infty}^{\infty} x[n-m]h[m] = \sum_{m=-\infty}^{\infty} e^{j\omega_0(n-m)}h[m] = e^{j\omega_0 n}\left(\sum_{m=-\infty}^{\infty} e^{-j\omega_0 m}h[m]\right) \\ &= e^{j\omega_0 n}H_d(\omega_0) = x[n]H_d(\omega_0) \end{aligned} \tag{3}$$

assuming that the DTFT $H_d(\omega)$ exists (i.e., the sum $H_d(\omega) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n}$ converges or, equivalently, $h[n]$ is finite for all n). Any input that gives $y[n] = Cx[n]$, where C is a constant (which may be complex), is considered an eigensequence.

2.2 Magnitude and Phase of the Frequency Response

Consider a known frequency response,

$$H_d(\omega) = |H_d(\omega)|e^{j\angle H_d(\omega)} \tag{4}$$

where $|H_d(\omega)|$ is the magnitude of the frequency response (sometimes referred to as the magnitude response of the LSI system) and $\angle H_d(\omega)$ is the phase of the frequency response (also referred to as the phase response or phase shift) of the system.

Recall that $H_d(\omega)$ is periodic with period 2π, i.e.

$$|H_d(\omega_0)| = |H_d(\omega_0 + 2\pi)| \tag{5}$$

$$\angle H_d(\omega_0) = \angle H_d(\omega_0 + 2\pi) \tag{6}$$

With that in mind, presented below are the three most common types of ideal filters. Only the magnitude spectrum is shown because the phase spectrum for an ideal filter is identically zero for all values of ω_0. This is because ideally the filter adds no delay at any frequency.

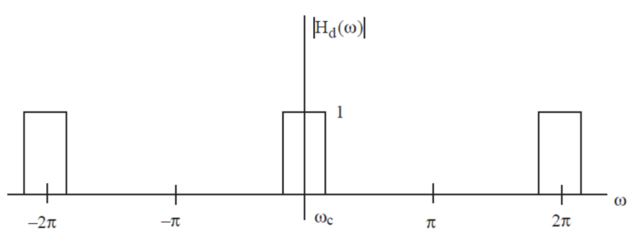


Figure 1: Ideal Lowpass Filter

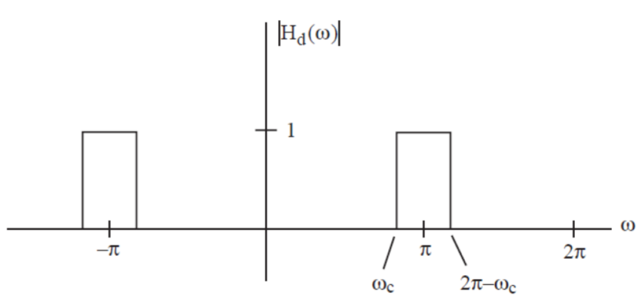


Figure 2: Ideal Highpass Filter

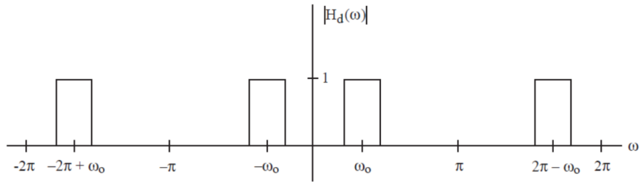


Figure 3: Ideal Bandpass Filter

For the above ideal filters, the corresponding impulse responses are

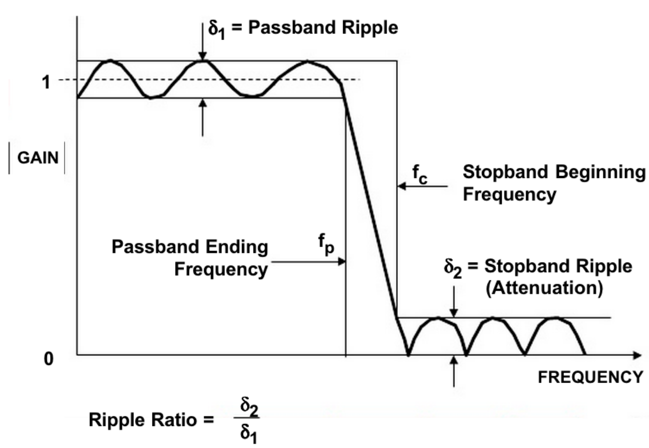
$$\text{Lowpass: } h[n] = \frac{\omega_c}{\pi}\text{sinc}(\omega_c n) \tag{7}$$

$$\text{Highpass: } h[n] = \delta[n] - \frac{\omega_c}{\pi}\text{sinc}(\omega_c n) \tag{8}$$

$$\text{Bandpass: } h[n] = \cos(\omega_0 n)\frac{\omega_c}{\pi}\text{sinc}(\omega_c n) \tag{9}$$

It is very important to note that these frequency response functions contain a sinc function, which is infinitely long and non-causal. Therefore, these ideal filters are unrealizable using a finite-order causal system (i.e., using any implementation that has a finite number of delay elements).

Instead, one can approximate the ideal filters with finite-order causal systems by shifting and windowing the sinc function in the time domain. The time-domain shift will result in the addition of a linear phase ('pure delay,' an equal delay across frequencies) in the frequency domain. The windowing will result in ripples ('Gibb's Phenomenon') in the passband and stopband and will result in a finite transition width. The magnitude frequency response of a realizable filter is described by the figure below



Note that although an equal ripple low-pass filter is used as an example, the characterization also applies to non-equal ripple filters.

A frequency response $H_d(\omega)$ can be written as

$$H_d(\omega) = |H_d(\omega)|e^{j\angle H_d(\omega)} \tag{10}$$

If the phase $\angle H_d(\omega)$ is of the form

$$\angle H_d(\omega) = \alpha\omega + \beta \tag{11}$$

where α and β are constants and α is real, then $H_d(\omega)$ is said to have Linear Phase.

In general, if $H_d(\omega)$ is Linear Phase, it can be expressed as

$$H_d(\omega) = R(\omega)e^{j(\beta+\alpha\omega)} \tag{12}$$

where $R(\omega)$ is a real-valued function of ω. Note that $R(\omega)$ can contribute $\pm\pi$ to the phase when $R(\omega) < 0$, because $-1 = e^{j\pi}$.

The process of adding linear phase to a system introduces a 'pure' delay to the overall processing of the system, which is often acceptable. Nonlinear phase adds frequency dependent delay, which can cause substantial distortion to a signal of interest, especially if the phase has a slope that changes dramatically over values of frequency in which the input signal has appreciable energy.

NOTE: When we refer to 'delay', we are typically interested in the 'group delay,' which is equal to the negative of the derivative of the phase, $\tau = -\frac{d}{d\omega}\angle H_d(\omega)$. If the phase has the form $\angle H_d(\omega) = -\alpha\omega + \beta$, then $\tau = \alpha$ is a constant! If not, then the delay τ could be a function of frequency, which can distort the siganl.

3 Lab Demo

3.1 Calculating the Frequency Reponse of an LSI System

Consider an infinite impulse response (IIR) lowpass filter

$$y[n] - 0.25838y[n-1] = x[n] + x[n-1] \tag{13}$$

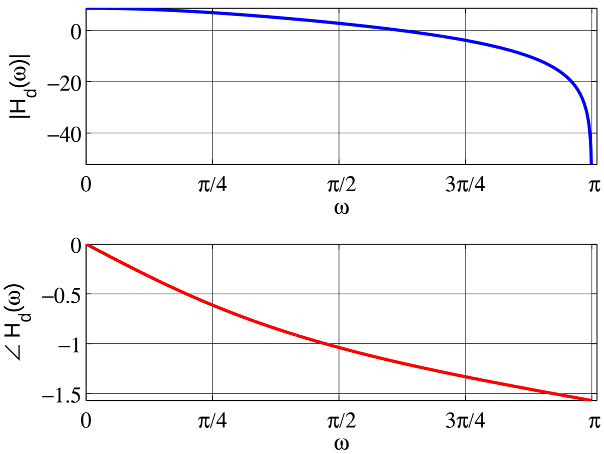
using the function `freqz`, the magnitude and phase of the frequency response can be visualized.

```
b = [1 1];
a = [1 -0.25838];
[h,w] = freqz(b,a,1024);

figure;
subplot(2,1,1);
plot(w,mag2db(abs(h)),'b','linewidth',2);
grid on;
set(gca,'GridLineStyle','-');
axis tight;
xlabel('\omega');
ylabel('|H_d(\omega)|');

subplot(2,1,2);
plot(w,angle(h),'r','linewidth',2);
grid on;
set(gca,'GridLineStyle','-');
axis tight;
xlabel('\omega');
ylabel('\angle H_d(\omega)');
```

The phase and magnitude response can also be plotted using the high level tool `fvtool`.

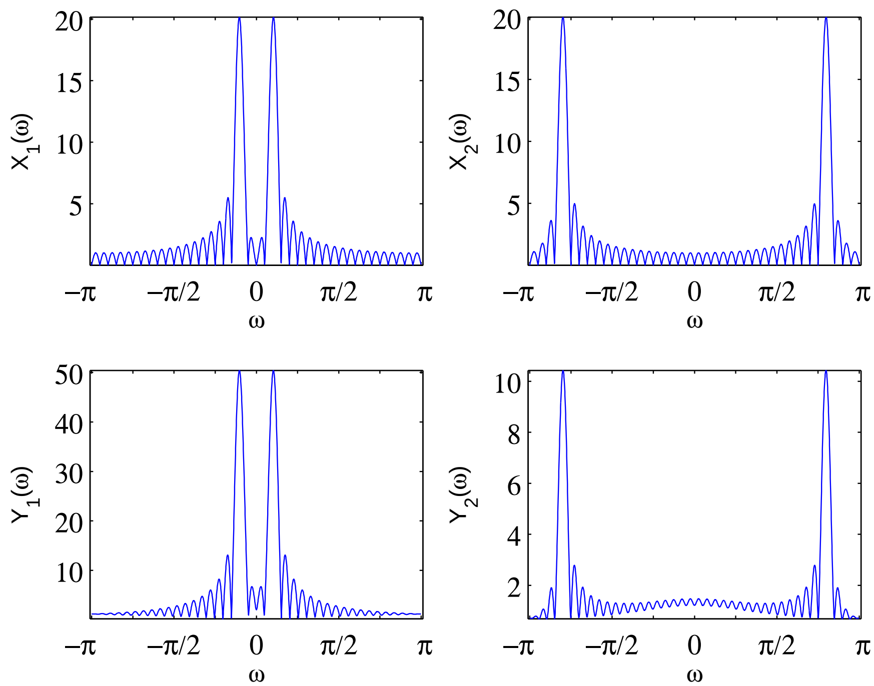


Now consider the two inputs (which are eigensequences, as described in section 2.1).

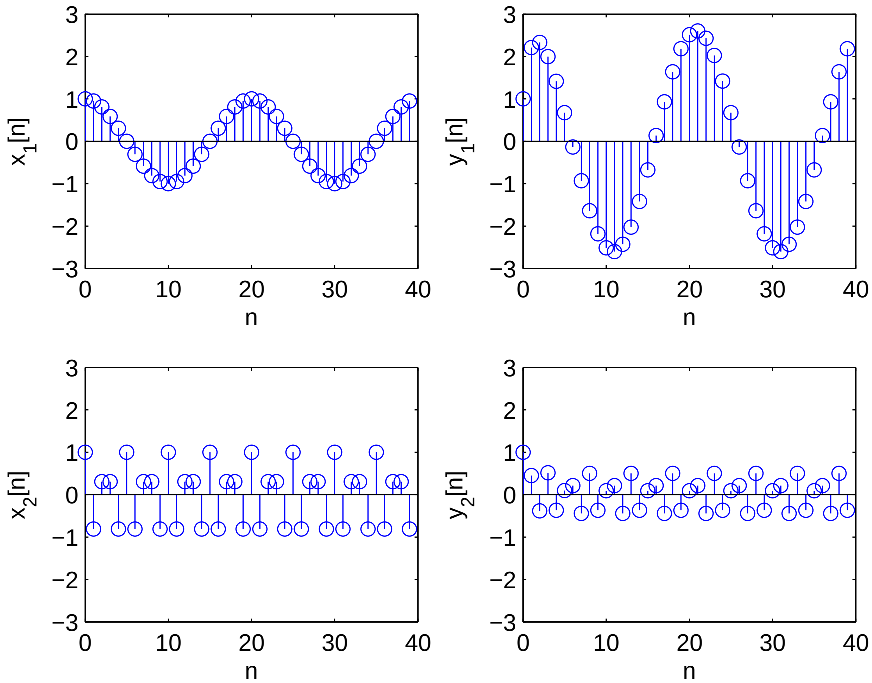
$$x_1[n] = \cos\left(\frac{\pi n}{10}\right) \tag{14}$$

$$x_2[n] = \cos\left(\frac{8\pi n}{10}\right) \tag{15}$$

After being passed through the system given by (13), the DFT spectra of $y_1[n]$ and $y_2[n]$ are as displayed below.



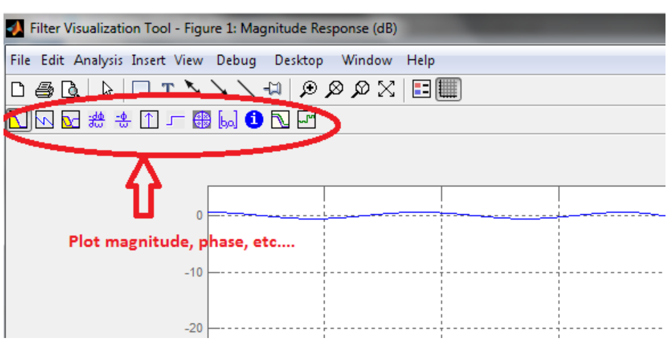
For $x_1[n]$, notice that the spectrum of $y_1[n]$ has a higher amplitude than the input. This is due to the frequency of $x_1[n]$ being in the passband of the filter, which has a gain larger than 1. However, for $x_2[n]$, the spectrum of $y_2[n]$ has been significantly attenuated. This is because the frequency of $x_2[n]$ is in the stopband of the filter. This can also be seen in the time-domain plots below.



3.2 Finite Impulse Response (FIR) Filters

Two lowpass FIR filters, designed with the Parks-McClellan method and the windowing method (using a Hamming window) are given in the figures below.

NOTE: *By "normalized cutoff frequency" we mean* ω_c/π. *Thus, if the DTFT frequency* ω *is in the range* $(-\pi, \pi)$, *the normalized values of* ω *fall in the range (-1,1). Also, the plots below are screenshots taken of* `fvtool`. *An unfortunate artifact of previous semesters is that the fonts are very small. Please zoom in on the pdf file to read these plots. You can recreate these figures yourself by first calculating the filter coefficients for each filter according to the specifications, then using* `fvtool` *to view the filter.*



The `fvtool` *GUI will display magnitude, phase, impulse response, pole-zero plot, and more!*

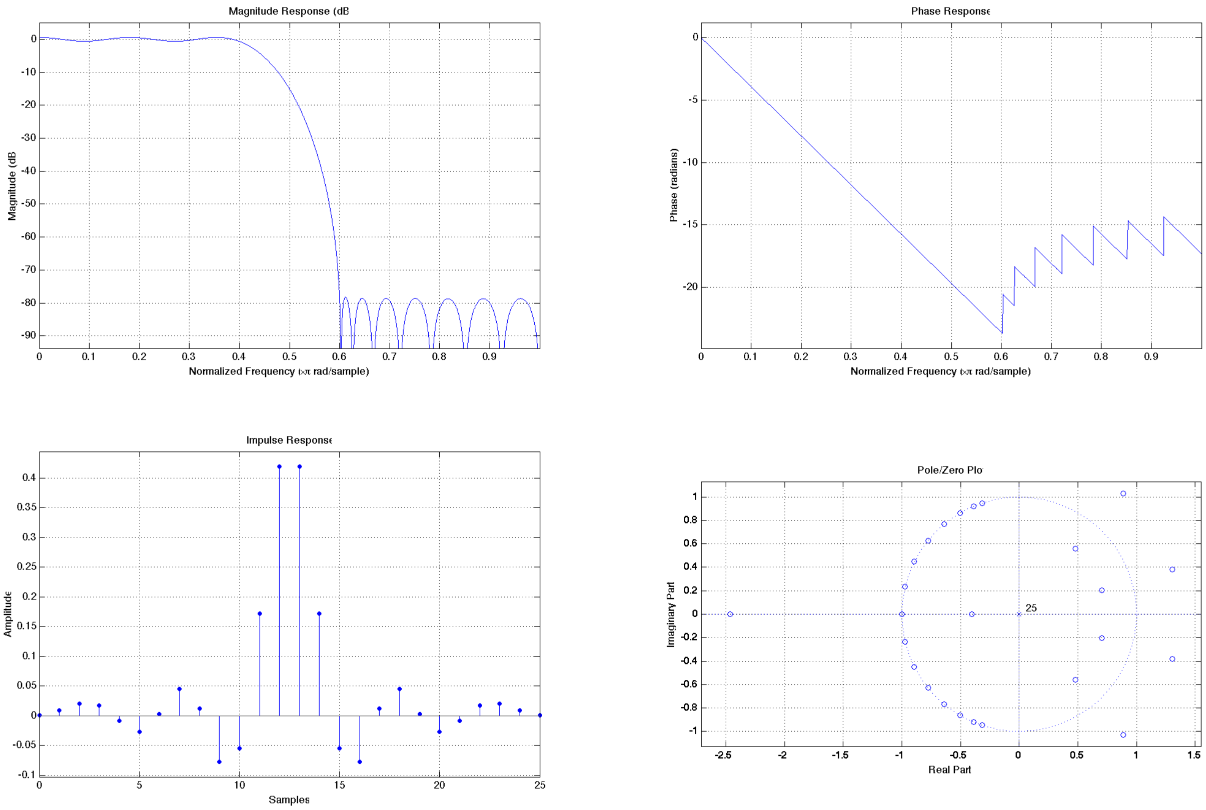


Figure 4: A lowpass FIR filter designed using the Parks-McClellan method. It is a filter of order 25 with a normalized passband cutoff frequency of 0.4, stopband cutoff frequency of 0.6, passband ripple 1 dB and stopband ripple 80 dB. Its frequency response, impulse response, and pole-zero plots are shown.

The Parks-McClellan filter may be calculated and shown as follows:

```
rp = 1;        % Passband ripple (in dB)
rs = 80;       % Stopband ripple (in dB)
f = [.4 .6];   % Normalized cutoff frequencies
a = [1 0];     % Desired amplitudes
dev = [(10^(rp/20)-1)/(10^(rp/20)+1) 10^(-rs/20)]; % Compute deviations (not in dB)
[n,fo,ao,w] = firpmord(f,a,dev); % Estimate the filter order -it turns out n = 25

b = firpm(n,fo,ao,w); % Design the filter
fvtool(b,1); % Display the filter
```

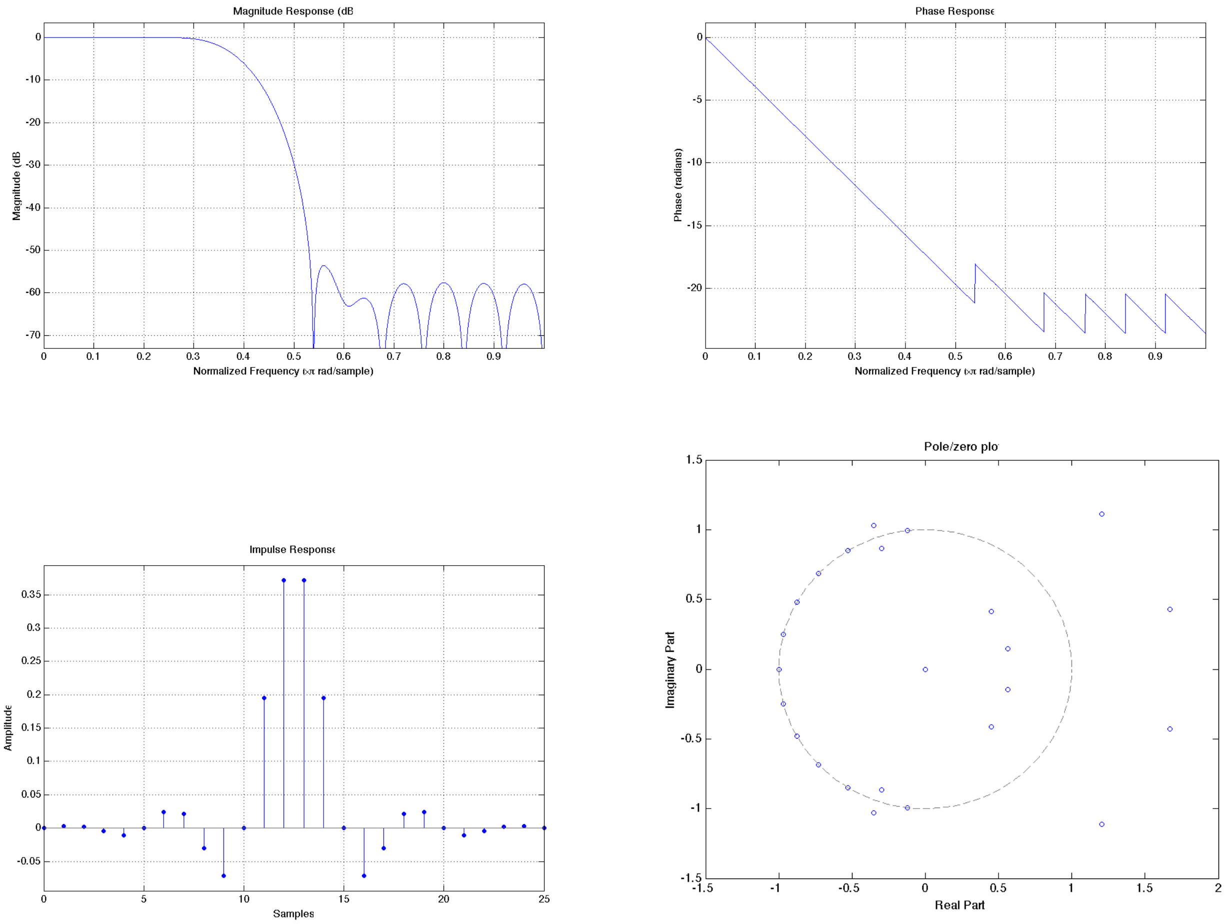


Figure 5: A lowpass FIR filter designed using the windowing method (with a Hamming window). It is a filter of order 25, with a passband cutoff frequency of 0.4. Its frequency response, impulse response, and pole-zero plot are shown.

Figure 6 shows the magnitude of the frequency responses of the above filters near the edges of the passband and stopband, which can be used to determine the ripples and cutoff frequencies of the passband and stopband.

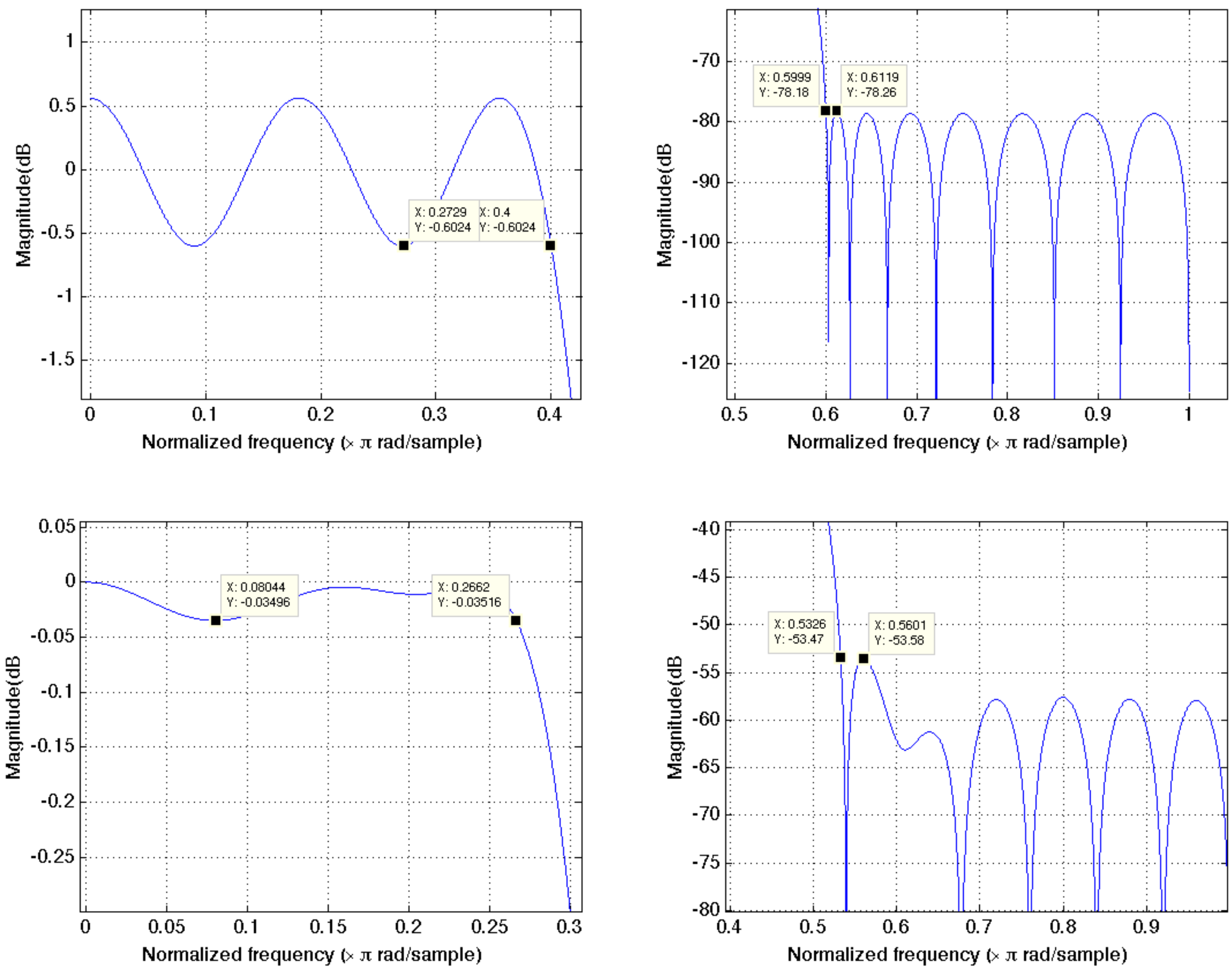


Figure 6: **(Top)** FIR filter designed by the PM algorithm: passband ripple = 0.6 dB, passband cutoff frequency = 0.4π, stopband ripple = 78.2 dB, stopband cutoff frequency = 0.6π. **(Bottom)** FIR filter designed by the windowing method: passband ripple = 0.017 dB, passband cutoff frequency = 0.27π, stopband ripple = 54.6 dB, stopband cutoff frequency = 0.53π. The width of the transition band is 0.26π. The cent is 0.4π. That is the cutoff frequency used in the design.

3.3 Infinite Impulse Response (IIR) Filters

Two lowpass IIR filters (an elliptical filter and a Butterworth filter) are given in the figures below.

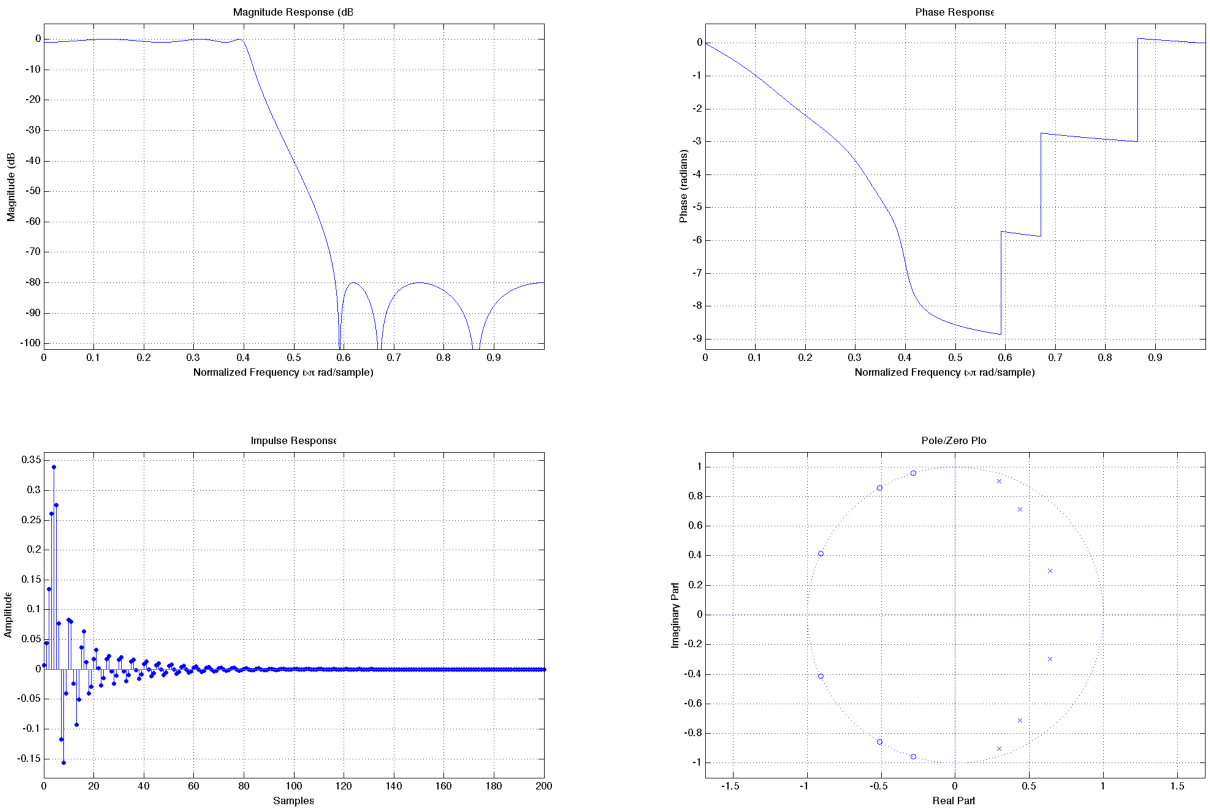


Figure 7: The frequency response of a lowpass elliptical IIR filter. It is of order 6, designed with a normalized passband cutoff frequency of 0.4, stopband cutoff frequency of 0.6, passband ripple of 1 dB and stopband ripple of 80 dB. Its frequency response, impulse response, and pole-zero plot are shown.

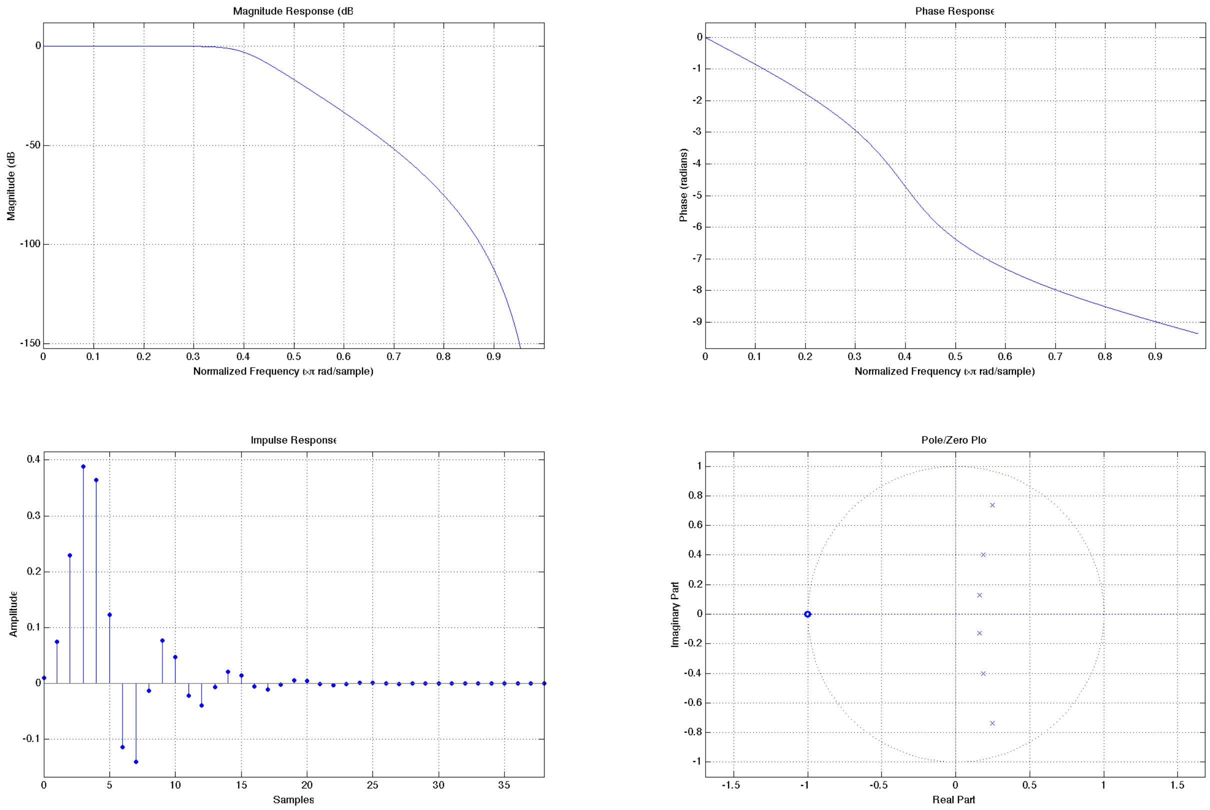


Figure 8: A lowpass Butterworth IIR filter. It is of order 16, designed with passband cutoff frequency 0.4. Its frequency response, impulse response, and pole-zero plot are shown.

Figure 9 shows the magnitude of the frequency responses of the above filters near the edges of the passband and stopband, which can be used to determine the ripples and cutoff frequencies of the passband and stopband.

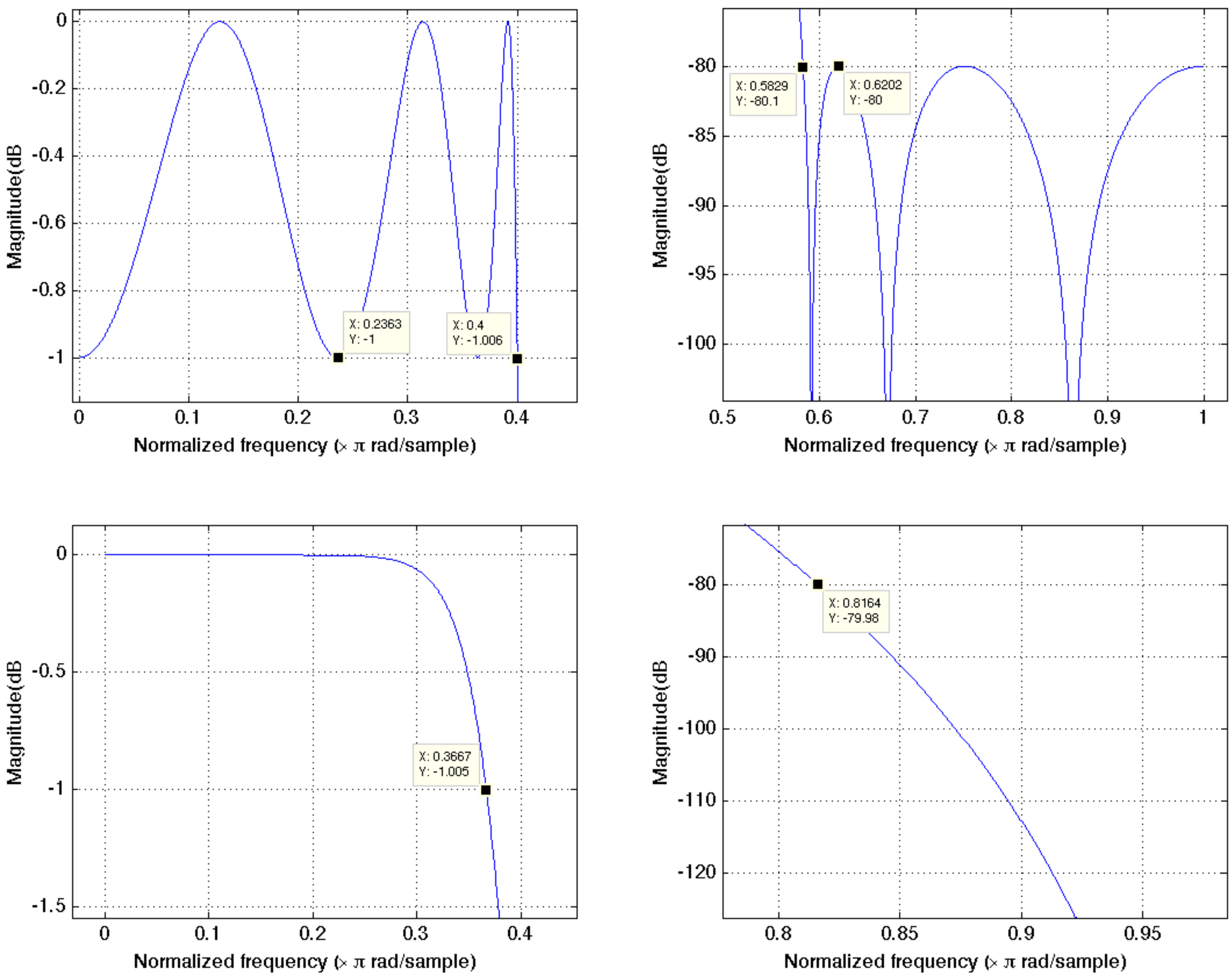


Figure 9: **(Top)** The elliptical IIR filter: passband ripple = 1 dB, passband cutoff frequency = 0.4π, stopband ripple = 80 dB, stopband cutoff frequency = 0.58π **(Bottom)** The Butterworth IIR filter: the magnitude response reaches 1 dB deviation in the passband at 0.37π, the magnitude response reaches 80 dB attenuation in the stopband at 0.82π.

3.4 Effects of Linear and Nonlinear Phase Filters

Consider two inputs

$$x_1[n] = \cos(0.2\pi n) \tag{16}$$
$$x_2[n] = \cos(0.4\pi n) \tag{17}$$

Both inputs are passed through FIR (Figure 5) and IIR filters (Figure 9, **top**) shown previously. The time-domain signal before (red) and after (blue or black) passing these filters are shown below. Notice the identical shift in the signals filtered by the FIR filter. In contrast, the IIR filter produces different shifts in the output signals.

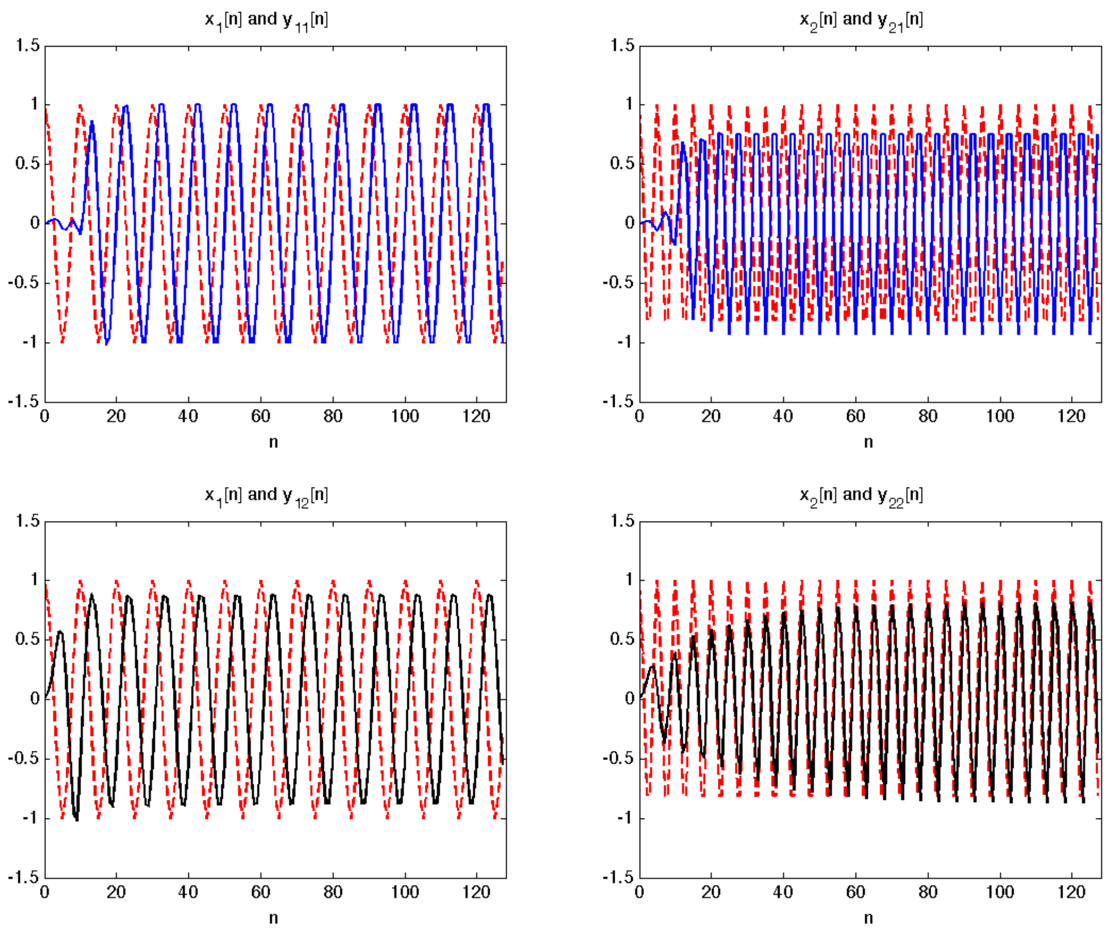


Figure 10: **(Top)** Signals that pass through the FIR filter **(Bottom)** Signals that pass through the IIR filter.

4 Homework 5 - Due 11/3/2015 at 5:00 PM

Many of the following problems will ask you to plot (i) the magnitude of the frequency response, (ii) the phase of the frequency response, (iii) the impulse response, and (iv) the pole-zero diagram. You may write your own code to make these plots, or use the `fvtool(b,a)` command in Matlab.

1. FIR Filter

   (a) Given the following high-pass FIR filter designed using the windowing method (Kaiser Window), calculate and plot the magnitude and phase of the frequency response, the impulse response, and the pole-zero diagram.

   ```
   Fs = 6000;
   d = fdesign.highpass('Fst,Fp,Ast,Ap',300,350,40,1,Fs);
   hd = design(d,'kaiserwin');
   b = hd.Numerator; a = 1;
   ```

   (b) Given the following high-pass FIR filter designed using the Parks-McClellan method, calculate and plot the magnitude and phase of the frequency response, the impulse response, and the pole-zero diagram.

   ```
   Fs = 6000;
   d = fdesign.highpass('Fst,Fp,Ast,Ap',300,350,40,1,Fs);
   hd = design(d,'equiripple');
   b = hd.Numerator; a = 1;
   ```

   (c) Compare the frequency responses of the filters in (a) and (b). What are the differences in the magnitude of the frequency responses? Is the phase of the frequency response linear or generalized linear? What are the passband and stopband edges? What are the ripple levels of the passband and stopband?

   (d) Given the following low-pass FIR filter designed using the Parks-McClellan method, calculate and plot the magnitude and phase of the frequency response, the impulse response, and the pole-zero diagram.

   ```
   Fs = 6000;
   d = fdesign.lowpass('Fp,Fst,Ap,Ast',900,950,1,40,Fs);
   hd = design(d,'equiripple');
   b = hd.Numerator; a = 1;
   ```

   (e) Given the following band-pass FIR filter designed using the Parks-McClellan method, calculate and plot the magnitude and phase of the frequency response, the impulse response, and the pole-zero diagram.

   ```
   Fs = 6000;
   d = fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2'
       ,500,550,650,700,40,1,40,Fs);
   hd = design(d,'equiripple');
   b = hd.Numerator; a = 1;
   ```

   (f) Load the signal 'flute2.wav'. This flute signal has been downsampled, so that $F_s = 6000$, to match the filters you just designed. Filter the audio using the filters in (b), (d) , and (e). Plot and compare the spectrograms of the signal before and after passing through these filters, and listen to the signals. What differences do you observe? *HINT: You can use code from Lab 4 to help you filter the signal and plot the spectrograms!*

2. IIR Filter

(a) Given the following elliptical high-pass IIR filter, calculate and plot the magnitude and phase of the frequency response, the impulse response, and the pole-zero diagram.

```
Fs = 6000;
d = fdesign.highpass('Fst,Fp,Ast,Ap',300,350,40,1,Fs);
hd = design(d,'ellip');
hd = convert(hd,'df1');
b = hd.coeffs.Numerator; a = hd.coeffs.Denominator;
```

(b) Compare the IIR filter in (a) with the FIR filter in Problem 1(b). What are the differences or similarities in the magnitude of the frequency responses? What are the differences or similarities in the phase of the frequency responses? What are the orders of the numerator and denominator polynomials?

(c) Given the following Chebyshev type II high-pass IIR filter, calculate and plot the magnitude and phase of the frequency response, the impulse response, and the pole-zero diagram.

```
Fs = 6000;
d = fdesign.highpass('Fst,Fp,Ast,Ap',300,350,40,1,Fs);
hd = design(d,'cheby2');
hd = convert(hd,'df1');
b = hd.coeffs.Numerator; a = hd.coeffs.Denominator;
```

(d) Compare the frequency of the filters in (a) and (c). What are the differences in the magnitude of the frequency responses? Are the phases of the frequency responses linear or generalized linear?

(e) Given the following elliptical low-pass IIR filter, calculate and plot the magnitude and phase of the frequency response, the impulse response, and the pole-zero diagram.

```
Fs = 6000;
d = fdesign.lowpass('Fp,Fst,Ap,Ast',900,950,1,40,Fs);
hd = design(d,'ellip');
hd = convert(hd,'df1');
b = hd.coeffs.Numerator; a = hd.coeffs.Denominator;
```

(f) Given the following elliptical band-pass IIR filter, calculate and plot the magnitude and phase of the frequency response, the impulse response, and the pole-zero diagram.

```
Fs = 6000;
d = fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2'
    ,500,550,650,700,40,1,40,Fs);
hd = design(d,'ellip');
hd = convert(hd,'df1');
b = hd.coeffs.Numerator; a = hd.coeffs.Denominator;
```

(g) For the filters in (a), (e), and (f), determine the passband and stopband edges. Determine the ripple levels of the passband and stopband.

(h) (*Not graded*) Compare your IIR filters to the FIR filters in problem 1 (e.g. compare the two high-pass filters, or two low-pass filters with the same pass- and stop-band parameters), using the 'flute2.wav' signal. Can you hear a faint distortion of the IIR filtered signals, compared to the FIR filtered signals? Distortion due to the IIR filters is not very noticeable here, in my opinion. Thus, we often use IIR filters even though there is distortion, since they have much lower order!