GEORGIA INSTITUTE OF TECHNOLOGY

SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2026 Fall 2013
Lab #11: Poles, Zeros and GUI for Filter Design

Date: 18-21 Nov. 2013

Each Lab assignment in ECE2026 consists of three parts: Pre-Lab, In-lab Tasks, and Take-home Questions. It requires you to come into lab prepared. Be sure to read the entire lab carefully before arriving.

Pre-Lab: You should read the Pre-Lab section of the lab and go over all exercises in this section before going to your assigned lab session. Although you do not need to turn in results from the Pre-Lab, doing the exercises therein will help make your in-lab experience more rewarding and go more smoothly with less frustration and panicking.

In-lab Tasks and Verification: There are a number of designated tasks for each student to accomplish during the lab session. Students are encouraged to read, prepare for and even attempt at these tasks beforehand. These tasks must be completed **during your assigned lab time** and the steps marked *Instructor Verification* must also be signed off **during the lab time**. One of the laboratory instructors must verify the appropriate steps by signing on the **Instructor Verification** line. When you have completed a step that requires verification, simply put a plastic cup on top of your PC and demonstrate the step to one of the TAs or the professor. (You can also use the plastic cups to indicate if you have a more general question, i.e. you can use it to get our attention even if you don't have an Instructor Verification ready.)

Take-home Questions: At the end of each lab sheet below all the verification steps, several questions are to be answered by the student, who can choose to complete the answers while in the lab or after leaving the lab.

The lab sheet with all verification signatures and answers to the questions needs to be turned in to the Lab-grading TA at the beginning of the next lab session.

1 Introduction

This lab introduces to you two GUIs that will help you understand the relationship between the frequency response and the placement of poles and zeros (a GUI named **`PeZ`**), and a basic methodology for designing a filter that satisfies some prescribed requirements (a GUI named **`filterdesign`**).

1.1 Frequency Response

In order to build an intuitive understanding of the relationship between the location of poles and zeros in the z-domain, the impulse response $h[n]$ in the n-domain, and the frequency response $H(e^{j\hat{\omega}})$ (the $\hat{\omega}$-domain), a graphical user interface (GUI) called **`PeZ`** was written in MATLAB for doing interactive explorations of the three domains.[1] **`PeZ`** is based on the system function, represented as a ratio of polynomials in z^{-1}, which can be expressed in either factored or expanded form as:

$$H(z) = \frac{B(z)}{A(z)} = G\frac{\prod_{k=1}^{M}(1-z_k z^{-1})}{\prod_{l=1}^{N}(1-p_l z^{-1})} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1-\sum_{l=0}^{N} a_l z^{-l}} \qquad (1)$$

where M is the number of zeros and N the number of poles. The poles are $\{p_l\}$, and the zeros are $\{z_k\}$. If the polynomials have real coefficients, complex poles and zeros must exist in conjugate pairs. Note: Due to the Thanksgiving holiday, the IIR filter concept in this lab is ahead of the lecture schedule (Intro to IIR takes place on Friday, 11/18/2013). However, the lab is entirely built upon earlier concepts of frequency response and system function, which you are already familiar with. The additional concept of feedback is not involved here.

[1]The original **`PeZ`** was written by Craig Ulmer; a later version by Koon Kong is the one that we will use in this lab.

The **`PeZ`** GUI is contained in the *SP-First* toolbox. To run **`PeZ`**, type **`pezdemo`** at the command prompt and you will see the GUI shown in Fig. 1.

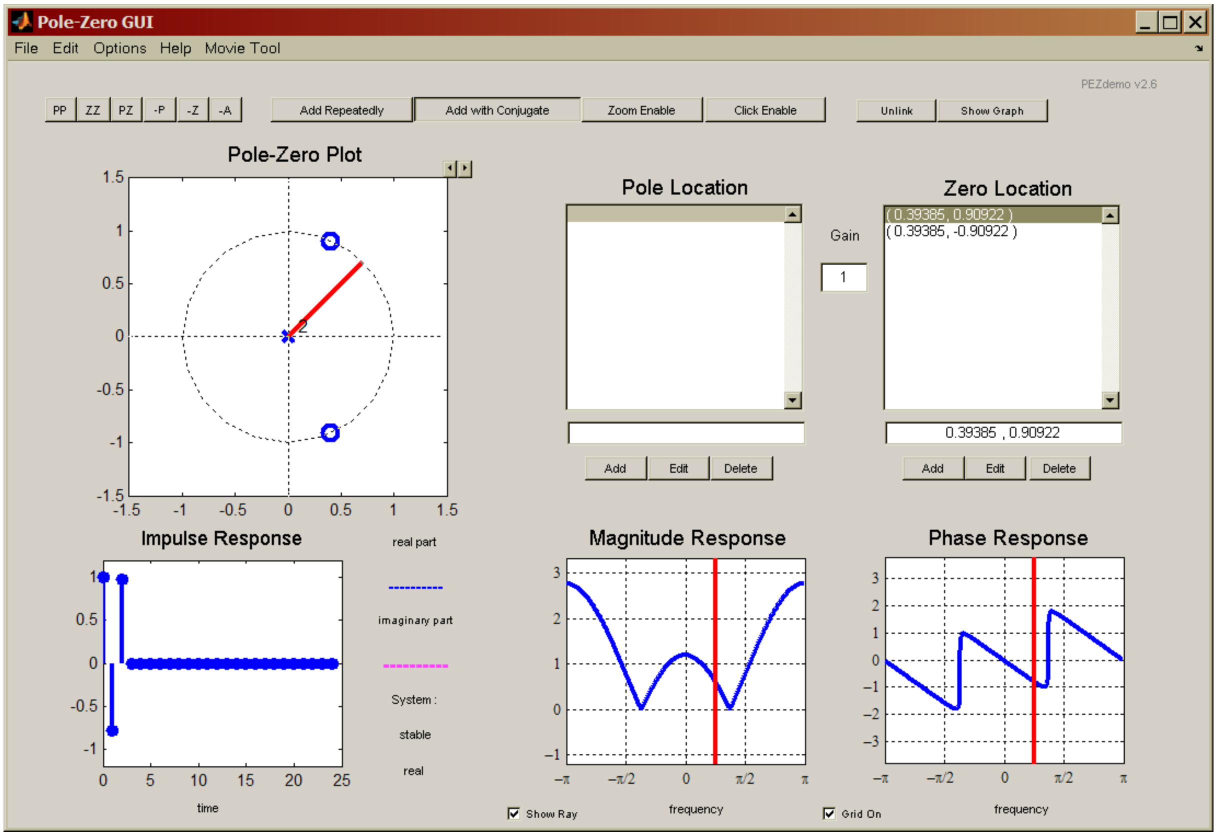

Figure 1: GUI interface for **`pezdemo`** running in MATLAB version 6 and later. A length-3 FIR filter is shown. Zero locations are given in rectangular coordinates.

1.2 Controls for `PeZ` using `pezdemo`

The **`PeZ`** GUI is controlled by the Pole-Zero Plot where the user can add (or delete) poles and zeros, as well as move them around with the pointing device. For example, Fig. 1 shows a case where two (complex-conjugate) zeros have been added, and **`PeZ`** has automatically included two poles at $z = 0$ so that the system is causal. The button denoted as **`ZZ`** was used to add these zeros; the button **`PP`** would be used to add poles. By default, the **`Add with Conjugate`** property is turned on, so poles and zeros are typically added in pairs to satisfy the complex-conjugate property, giving real coefficients.

To learn about other controls in **`pezdemo`**, access the menu item called "**`Help`**" for extensive information about all the **`PeZ`** controls and menus. Here are a few things to explore. In the **`Pole-Zero Plot`** panel you can selectively place poles and zeros in the z-plane, and then observe (in the other plotting panels) how their placement affects the impulse and frequency responses. The red ray in the z-domain window is tied to the red vertical lines on the frequency responses, and they move together. This helps identify frequency domain features that are caused by pole locations or zero locations, because the angle around the unit circle corresponds to frequency $\hat{\omega}$. In **`PeZ`** an individual pole/zero pair can be moved around and the corresponding $H(e^{j\hat{\omega}})$ and $h[n]$ plots will be updated as you drag the pole (or zero). Since exact placement of poles and zeros with the mouse is difficult, an **`Edit`** button is provided for numerical entry of the real and imaginary parts. Before you can edit a pole or zero, however, you must first select it in the list of **`Pole Locations`** or **`Zero Locations`**. Next, you will have to make the necessary change in the edit field (box above the **`Edit`** button) and then press the

$\boxed{\texttt{Edit}}$ button for the change to take effect. Removal of individual poles or zeros can also be performed by using the $\boxed{\texttt{-P}}$ or $\boxed{\texttt{-Z}}$ buttons, or with the $\boxed{\texttt{Delete}}$ button. Note that all poles and/or zeros can be easily cleared by clicking on the $\boxed{\texttt{-A}}$ button.

1.3 Filter Design by Windowing

Here we introduce the design of FIR filters to produce good frequency responses for low-pass, band-pass and high-pass filter specifications. The MATLAB GUI **`filterdesign`** will be used to do the filter designs by a method called "windowing." Once you have the filter, you can use **`firfilt()`**, or **`conv()`**, to implement filters and **`freqz()`** to verify the filter's frequency response.[2] As a result, you can learn how a filter operates by knowing how it treats different frequency components in the input. Band-pass filters can be used to detect and extract information from sinusoidal signals, e.g., individual notes in a musical passage or tones in a touch-tone telephone dialer (in the previous lab).

For the process of filter design, it is important to recognize that the choice of filter coefficients $\{b_k\}$ determines the frequency response. The most useful filter design methods provide a formula for the $\{b_k\}$ coefficients that will give a particular behavior for $H(e^{j\hat{\omega}})$, e.g., when the filter coefficients are given by the formula:

$$h[n] = \frac{\sin(\hat{\omega}_c(n - M/2))}{\pi(n - M/2)}(0.54 - 0.46\cos(2\pi n/M)) \qquad \text{for } n = 0, 1, 2, \cdots, M \tag{2}$$

where M is the filter order, which should be an **even integer**. The first term in $h[n]$ is a ***sinc function***; the second term in $h[n]$ is called a Hamming window. The design parameter $\hat{\omega}_c$ is called the ***cutoff frequency*** of the filter because it determines the pass-band and stop-band regions of the frequency response (see Fig. 3 below). The following MATLAB statements show how to use **`freqz`** to compute and plot both the magnitude (absolute value) and the phase of the frequency response of the filter in (2) when $\hat{\omega}_c = 0.32\pi$. The frequency response is plotted as a function of $\hat{\omega}$ in the range $-\pi \le \hat{\omega} \le \pi$:

```
M = 50; nn = 0:M;
wc = 0.32*pi;
sincwc = sin(wc*(nn-M/2))./(pi*(nn-M/2)); %-- sinc function
sincwc(M/2 + 1) = wc/pi; %-- fix divide by zero
HW = 0.54 - 0.46*cos(2*pi*nn/M); %-- Hamming window
num = sincwc.*HW; %-- Filter Coefficients in numerator of z-transform
den = 1; %-- denominator is 1 for FIR filter
%----------------------------------------------------------
%--- Use filter coefficients to make frequency response plots
%-------
ww = 0:(pi/2000):pi; %-- omega hat
HH = freqz(num, den, ww);        %<--freekz.m is an alternative
subplot(2,1,1);
plot(ww, abs(HH)), zoom on, grid on
subplot(2,1,2);
plot(ww, angle(HH)), zoom on, grid on, shg
xlabel('Normalized Radian Frequency')
```

For FIR filters, the second argument of **`freqz( , 1, )`** must always be equal to 1 (i.e., zeroth order). The frequency vector **`ww`** should cover an interval of length π or 2π for $\hat{\omega}$, and its spacing must be fine enough to give a smooth curve for the plots[3] of $H(e^{j\hat{\omega}})$.

2 Pre-Lab

[2]If you do not have the function **`freqz.m`**, there is a substitute available in the *SP-First Toolbox* called **`freekz.m`**.

[3]If the output of the **`freqz`** function is not assigned, then plots are generated automatically; however, the magnitude is given in decibels which is a logarithmic scale. For linear magnitude plots a separate call to plot is necessary as in the code above.

2.1 Using `PeZdemo` GUI

2.1.1 Create an FIR Filter with `PeZ`

Implement the following FIR system:

$$H(z) = 1 - z^{-1} + z^{-2}$$

by factoring the polynomial and placing the two zeros correctly. Observe the following two facts:

- The impulse response $h[n]$ values are equal to the polynomial coefficients of $H(z)$.
- The frequency response has nulls because the zeros of $H(z)$ are exactly on the unit circle. Compare the frequencies of the nulls to the angles of the zeros.

Move the zero-pair around the unit circle and observe that the location of the null also moves.

2.1.2 Create an IIR Filter with `PeZ`

Implement the following first-order IIR system:

$$H(z) = \frac{1 - z^{-1}}{1 + 0.9z^{-1}} = \frac{z - 1}{z - (-0.9)}$$

by placing its pole and zero at the correct locations in the z-plane. First, try placing the pole and zero with the mouse, and then use the **`Edit`** feature to get exact locations. Since **`PeZ`** wants to add complex-conjugate pairs, you should turn off the **`Add with Conjugate`** feature to add a single pole. Look at the frequency response and determine what kind of filter you have.

Now, use the mouse to "grab" the pole and move it from $z = -0.9$ to $z = +0.8$. To move along the real axis, you can use **`Options -> Move on Real Line`** from the GUI menu. Observe how the frequency response changes. Describe the type of filter that you have created (i.e., HPF, LPF, or BPF).

2.2 Using `filterdesign` GUI

You can find GUI **`filterdesign`** in the *SP-First* toolbox. The interface is shown in Fig. 2. Both FIR and IIR filters can be designed, but we will only be working in the FIR case which would be selected with the radio button in the upper right. (You may find the IIR part useful in the future though.) Once FIR is selected, the window type should be selected from the drop-down list in the lower right. Lastly, it is necessary to set the order of the FIR filter and the cutoff frequency; these parameters can be entered in the edit boxes. For practice, try to carry out the same FIR filter design as in the MATLAB code in Section 1.3. M =50 and $\hat{\omega}_c = 0.32\pi$.

2.2.1 Passband Defined for the Frequency Response

Certain types of digital filters have a frequency response (magnitude) that is close to one in some frequency regions, and close to zero in others. For example, the plot in Fig. 3 is a low-pass filter whose magnitude is close to one when the frequency $\hat{\omega}$ is near zero. This region is called the ***pass-band*** of the filter. It will be useful to have a precise measurement of the pass-band width so that we can compare different filters.

(a) From the plot of the magnitude response in Section 1.3 determine the set of frequencies where the magnitude is very close to one, as defined by $\big| |H(e^{j\hat{\omega}})| - 1 \big|$ being less than 0.01. This should be a region of the form $-\hat{\omega}_p \le \hat{\omega} \le \hat{\omega}_p$. Determine $\hat{\omega}_p$ for the case where M =50 and $\hat{\omega}_c = 0.32\pi$.

(b) The parameter $\hat{\omega}_p$ is called the *pass-band edge*. Compare the value of $\hat{\omega}_p$ found in the previous part to the design parameter $\hat{\omega}_c$ in (2).

2.2.2 Stop-band Defined for the Frequency Response

When the frequency response (magnitude) of the digital filter is close to zero, we have the stop-band region of the filter. In the low-pass filter example of Fig. 3, the magnitude is close to zero when the frequency $\hat{\omega}$ is near π (a high frequency). This region is called the ***stop-band*** of the filter. We can make a precise measurement of the stopband edge as follows:

(a) From the plot of the magnitude response that you made in Section 1.3, or with the **`filterdesign`** GUI, determine the set of frequencies where the magnitude is nearly zero, as defined by $|H(e^{j\hat{\omega}})|$ being less than 0.01. In decibels, 0.01 is –40 dB. This should be two regions: $\hat{\omega}_s \le \hat{\omega} \le \pi$ in positive frequencies, and $-\pi \le \hat{\omega} \le -\hat{\omega}_s$ for negative frequencies. Determine $\hat{\omega}_s$ for the case where $M = 50$ and $\hat{\omega}_c = 0.32\pi$.

(b) The parameter $\hat{\omega}_s$ is called the *stop-band edge*. Compare the value of $\hat{\omega}_s$ found in the previous part, and the value of $\hat{\omega}_p$ from Section 2.2.1 part (a), to the design parameter $\hat{\omega}_c$ in (2).

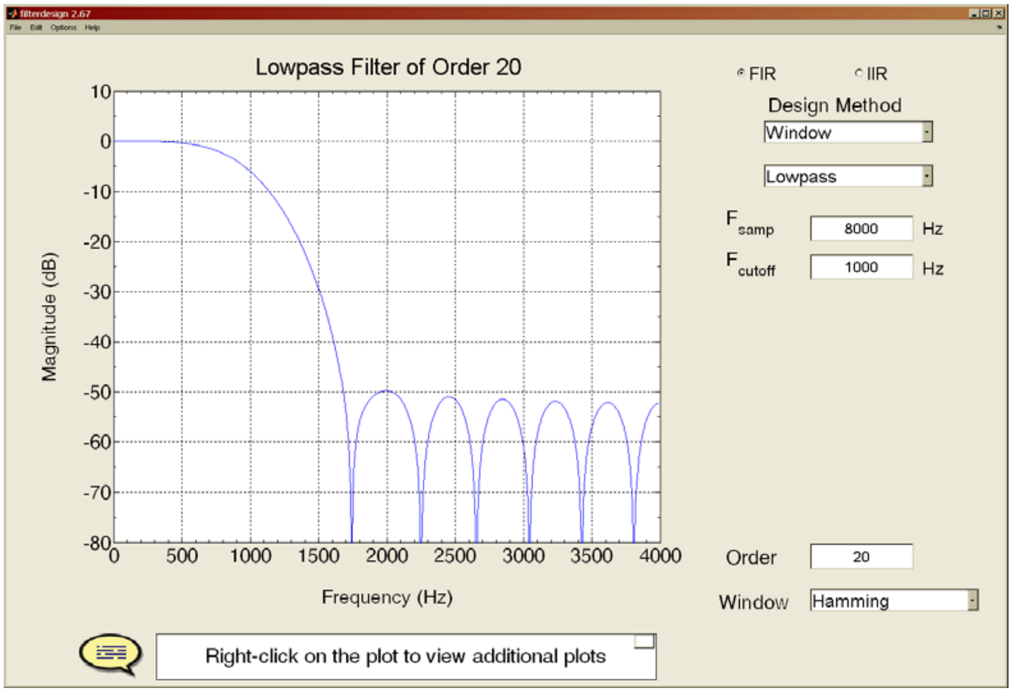

Figure 2 Filter design demo interface. The frequency is Hz, but can be displayed as $\hat{\omega}$ by clicking on the "Frequency" label. The amplitude scale is linear, but can be changed to a log scale by clicking on the "Magnitude" label. The log scale is in decibels, $20\log_{10}(A)$. When the Filter Choice is set to FIR Filters, many different window types can be selected, including the Hamming window.

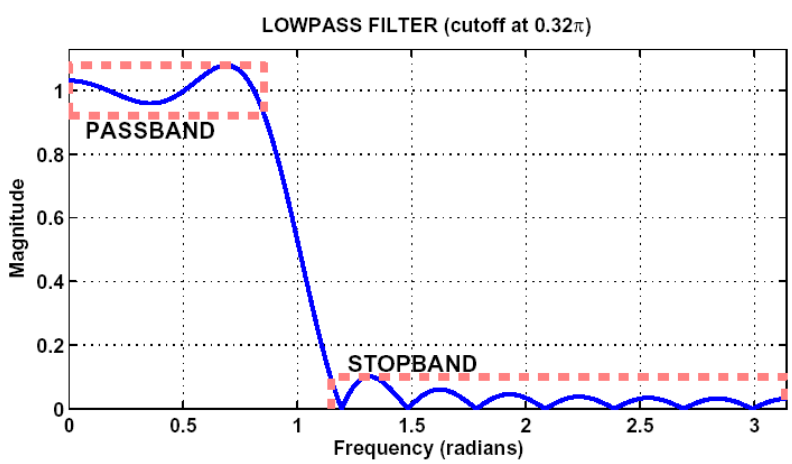

Figure 3 Pass-band and stop-band defined for a typical low-pass filter. This particular filter is a length-21 FIR filter designed with a rectangular window and a sinc function with a cutoff frequency of $\hat{\omega}_c = 0.32\pi$. The approximate value of the pass-band edge is $\hat{\omega}_p = 0.27\pi = 0.85$; the stop-band edge, $\hat{\omega}_s = 0.366\pi = 1.15$. The pass-band ripple is approximately ±0.1, and the stop-band ripple is also about ±0.1. The stop-band ripples are all positive on the magnitude plot.

2.2.3 Linear Phase in the Frequency Response

The phase of the frequency response can be related to time delay. In the low-pass filter example of Section 1.3, the phase plot appears to be jagged, but it is actually (piecewise) linear.

(a) Determine the slope of the linear segments of the frequency response for the $M = 50$ filter in Section 1.3. Your answer should be an integer.

(b) Plot the impulse response of the digital filter in Section 1.3 for the range $n = 0, 1, \ldots$, M. Then determine the symmetry point for $h[n]$, i.e., find the integer n_s such that $h[n_s + n] = h[n_s - n]$.

(c) Compare the value of the slope (from part (a)) to the "symmetry point" of $h[n]$. Verify that the phase slope is equal to $-n_s$.

3 Lab Exercise

The lab verification requires that you write down your observations on the verification sheet when using the GUIs **(`PeZdemo`** ***and*** **`filterdesign`).** ***These written observations will be graded.***

3.1 Relationships between z, n, and $\hat{\omega}$ domains – use of `PeZdemo` GUI

Work through the following exercises and keep track of your observations by filling in the worksheet at the end of this assignment. In general, you want to make note of the following quantities:

- Is the length of $h[n]$ finite or infinite? For the FIR case, what is the total length of the filter?
- How does $h[n]$ change with respect to its rate of decay for IIR filters? For example, when the impulse response is of the form $h[n] = a^n u[n]$, the impulse response will fall off more rapidly when a is smaller; and if $|a| > 1$ the impulse response blows up.
- If $h[n]$ exhibits an oscillating component, what is the period of oscillation? Also, estimate the decay rate of the "envelope" that overlays the oscillation.
- How does $H(e^{j\hat{\omega}})$ change with respect to null location, peak location and/or peak width?

Note: review the "Three-Domains - FIR" under the Demos link for Chapter 7 and "Three-Domains - IIR" under the Demos link for Chapter 8 for movies and examples of these relationships.

3.1.1 Real Poles (Optional)

`PeZ` allows you to place poles and zeros at various locations and let you observe the system's impulse response and frequency response. At the time of this lab, we have just started introducing the IIR system, to which the locations of poles are particularly important. Since the notion of IIR is rather new to the class, we make this subsection 3.1.1 an optional exercise. You may go to 3.1.2 directly. However, it is strongly recommended that once you understand IIR better, you come back and do the exercise in this section at your own pace.

(a) Use **`PeZ`** to place a single pole at $z = -3/4$. You may have to use the **`Edit`** button to get the location exactly right. Describe the important features of the impulse response $h[n]$.

(b) Also, describe the important features of the frequency response $H(e^{j\hat{\omega}})$, e.g., peak location and width. Furthermore, use the plots in **`PeZ`** for this case as the reference for answering the next three parts.

(c) Move the pole slowly from a location close to the origin ($z = 0$) out to $z = -3/4$, and then out to $z = -0.999$. Stay on the real axis by using **`Options -> Move on Real Line`** from the GUI menu. Observe the changes in the impulse response $h[n]$ and the frequency response $H(e^{j\hat{\omega}})$. Record your observations on the Verification Sheet.

Note: When you move poles and zeros, the impulse response and frequency response plots are updated continually in **`PeZ`**. Select the pole you want to move and start to drag it slowly. At the same time, watch for the update of the plots in the impulse response and frequency response panels.

(d) Place the pole exactly on the unit circle (or maybe just inside at a radius of 0.99999999). Describe the changes in $h[n]$ and $H(e^{j\hat{\omega}})$. What do you expect to see for $H(e^{j\hat{\omega}})$?

(e) Now, move the pole outside the unit circle. Describe the changes in $h[n]$. Explain how the appearance of $h[n]$ validates the statement that the system is not stable.

Note: By stability we mean that the system's output does not blow up as $n \to \infty$. In the unstable case, the frequency response $H(e^{j\hat{\omega}})$ is not legitimate because the system is no longer stable.

(f) In general, where should the poles be placed to guarantee system stability?

Instructor Verification (optional)

3.1.2 Zeros

A 6-point FIR digital filter has the following system function:

$$H(z) = \sum_{n=0}^{5} (-1)^n z^{-n} = \frac{1 - z^{-6}}{1 + z^{-1}}$$

(a) Use the roots function in MATLAB to determine the zeros of $H(z)$ for this FIR filter. In the rational form, the numerator has six zeros, and the denominator one pole, but there is a pole-zero cancellation.

(b) Use **`PeZ`** to place the five zeros of the FIR filter at the correct locations; use the **`edit`** feature to enter the exact value of the zeros. Observe that the impulse response will be ones and minus ones when you have the correct zero locations. Describe the frequency response of the filter: low-pass vs. high-pass.

(c) List the frequencies $(\hat{\omega})$ that are nulled by the length-6 FIR filter.

(d) One of the zeros will be at $z = +1$. Take that zero and move it from $z = +1$ to $z = -1$. Write down the system function $H(z)$ for the new filter.

Hint: The GUI contains the answer if you know where to look.

(e) Describe the frequency response of the filter created in part (d), i.e., LPF, HPF, or BPF.

(f) Now make a new filter that nulls one sinusoid, $x[n] = \cos(0.35\pi n)$ by placing one pair of zeros correctly. Determine the filter coefficients for this filter.

Note: it is not possible to place these zeros precisely enough with the GUI, so you will have to do some calculations to get the real and imaginary parts of the zeros.

Instructor Verification (separate page)

3.2 Filter Design (`filterdesign`) GUI

In this section you will experiment with different values for the parameters of the Hamming window to see the effect on the frequency response of the FIR filter (Sections 1.3 and 2.2).

(a) Start the **`filterdesign`** GUI[4], and select[5] a set of nominal parameters for a low-pass filter: $\hat{\omega}_c = 0.3\pi$, and $M = 40$. Observe the following parameters of the frequency response so that comparisons can be

[4] The **`filterdesign`** and **`dltidemo`** GUIs are part of the *SP-First* toolbox, which is already installed in the ECE lab. The latest versions of all the *SP-First* GUIs are included in the *SP-First* toolbox which can be found at: http://users.ece.gatech.edu/ mcclella/SPFirst/Updates/SPFirstMATLAB.html

[5] Click on the "frequency" label to toggle the units of the frequency axis between $\hat{\omega}$ and f in hertz.

made in the following parts: pass-band edge $(\hat{\omega}_p)$, stop-band edge $(\hat{\omega}_s)$, pass-band ripple $(\pm\delta_p)$, and stop-band ripple $(\pm\delta_s)$. It is difficult to read precise values from the GUI, so **use approximate values**.

Hint: Use **`Options->Grid and Options->Zoom`** while measuring.

Note: For the Hamming-window filter, the ripples should be less than ±0.01.

(b) Double the filter order, *M*, to 80. Describe how the frequency response changes. Determine the **approximate values** of the new band edges and ripples.

Note: The band edges lie on either side the cutoff frequency, $\hat{\omega}_c$.

(c) Halve the filter order, *M*, i.e., *M* = 20. Determine the **approximate values** of the new band edges and ripples.

(d) With *M* = 40, change the value of the cutoff frequency $\hat{\omega}_c$. Describe what happens when $\hat{\omega}_c$ is increased to 0.75π. Determine the **approximate values** of the new band edges and ripples.

Instructor Verification (separate page)

3.3 Estimating the Filter Order

The interval between stop-band edge and the pass-band edge is called the transition region; its width is called the *transition width*: $\Delta\hat{\omega} = |\hat{\omega}_s - \hat{\omega}_p|$. It can be used to estimate the filter order *M*.

3.3.1 Export the Filter Coefficients

It is difficult to make precise measurements in the **`filterdesign`** GUI. However, the **`filterdesign`** GUI has the capability to export the filter coefficients to the workspace, under the menu **`File->ExportCoeffs`**. The default for the export is to create two vectors called **`num`** and **`den`** which correspond to the numerator and denominator polynomials in the *z*-transform $H(z) = B(z)/A(z)$. For FIR filters the denominator $A(z)$ is always 1. For the numerator, the num vector contains the coefficients of $B(z)$ which are also the filter coefficients. Once the filter coefficients are saved in the workspace, the frequency response can be displayed in MATLAB using part of the code from Section 1.3.

(a) For the *M* = 40 filter that you designed in Section 3.2, determine precise values for the band-edges. Three significant digits should provide sufficient accuracy. These should refine your observations from Section 3.2(a).

(b) Repeat the precision band edge measurements for the *M* = 80 and *M* = 20 filters in Section 3.2.

3.3.2 Formula to Predict Filter Order

Use the band edge measurements from the previous section to derive a formula for the dependence of filter order on $\Delta\hat{\omega}$ as *M* is increased or decreased. The formula is only approximate, but it should capture the essential relationship between *M* and $\Delta\hat{\omega}$.

Hint: Look for a simple direct or inverse proportional formula.

Note: The filter order *M* should have little or no dependence on the ripples.

Instructor Verification (separate page)

Lab #11
ECE-2026 Fall-2013
Lab Sheet (2 pages)

Name: ____ Date of Lab:_______________

3.1.1 is optional

Part	**Observations**
3.1.1(a)	Describe impulse response
3.1.1(b)	Describe frequency response
3.1.1(c)	Move pole (0 → –0.9999); describe changes in $h[n]$ and $H(e^{j\hat{\omega}})$
3.1.1(d)	Pole on unit circle; describe changes in $h[n]$ and $H(e^{j\hat{\omega}})$
3.1.1(e)	Pole outside unit circle; describe changes in $h[n]$
3.1.1(f)	Requirement on poles to guarantee stability.

Verified:_________________________ Date/Time:_____________________________

3.1.2(a)	Zeros of length-6 FIR filter
3.1.2(b)	Describe Frequency Response of Filter in part (a)
3.1.2(c)	List frequencies that are nulled
3.1.2(d)	After moving the zero from +1 to –1, $H[z] =$
3.1.2(e)	Describe Frequency Response of Filter in part (d)
3.1.2(f)	Filter Coefficients of Nulling Filter:

Verified:_________________________ Date/Time:_____________________________

Part	Observations
3.2(a)	LPF: $\hat{\omega}_c = 0.3\pi$, and $M = 40$: band edges: $(\hat{\omega}_p)$, $(\hat{\omega}_s)$, and ripples: $(\pm\delta_p)$, $(\pm\delta_s)$.
3.2(b)	LPF: $\hat{\omega}_c = 0.3\pi$, and $M = 80$: band edges: $(\hat{\omega}_p)$, $(\hat{\omega}_s)$, and ripples: $(\pm\delta_p)$, $(\pm\delta_s)$.
3.2(c)	LPF: $\hat{\omega}_c = 0.3\pi$, and $M = 20$: band edges: $(\hat{\omega}_p)$, $(\hat{\omega}_s)$, and ripples: $(\pm\delta_p)$, $(\pm\delta_s)$.
3.2(d)	LPF: $\hat{\omega}_c = 0.75\pi$, and $M = 40$: band edges: $(\hat{\omega}_p)$, $(\hat{\omega}_s)$, and ripples: $(\pm\delta_p)$, $(\pm\delta_s)$.

Verified:___________________________ Date/Time:_______________________________

Part	Observations
3.3.1(a)	LPF: $\hat{\omega}_c = 0.3\pi$, and $M = 40$: **precise** band edges: $(\hat{\omega}_p)$, $(\hat{\omega}_s)$. Also: $\Delta\hat{\omega} =$
3.3.1(b)	LPF: $\hat{\omega}_c = 0.3\pi$, and $M = 80$: **precise** band edges: $(\hat{\omega}_p)$, $(\hat{\omega}_s)$. Also: $\Delta\hat{\omega} =$
3.3.1(b)	LPF: $\hat{\omega}_c = 0.3\pi$, and $M = 20$: **precise** band edges: $(\hat{\omega}_p)$, $(\hat{\omega}_s)$. Also: $\Delta\hat{\omega} =$
3.3.2	Formula for Filter Order :

Verified:___________________________ Date/Time:_______________________________