ECG SIGNAL FILTERING 6

Electrical Engineering 20N
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

HSIN-I LIU, JONATHAN KOTKER, HOWARD LEI, AND BABAK AYAZIFAR

1 Introduction

In this lab session, we will use LabVIEW to explore a practical application of the discrete-time filters that we have been studying in lectures, lab sessions, and discussion sections. Used in the medical fields, an **electrocardiogram (ECG)** is generated by an electrocardiograph, which measures electrical activity in the heart. ECGs are highly useful in studying heart behavior and in diagnosing potential heart problems. A common and well-documented problem in generating ECGs, however, is **power line interference**, which has a frequency of 60 Hz and arises due to the power lines connected to an electrocardiograph. There are other kinds of interferences in generating ECGs, such as white noise, but we shall only concern ourselves with the power line interference and white noise in the course of this laboratory session. Since power line interference occurs at a frequency of 60 Hz, a good choice to filter out this interference would be a notch filter, whose notch occurs at that frequency. We will use LabVIEW to simulate an ECG signal with the 60 Hz interference and operate upon that signal with a notch filter, and we will design this filter ourselves from scratch.

1.1 Lab Goals

- Implement a practical application of discrete-time filters in LabVIEW.
- Employ geometric reasoning to design a filter with the proper parameters to satisfy the required specifications.
- Analyze the magnitude and the phase of the frequency responses of practical discrete-time filters, and use LabVIEW to verify this analysis.
- Get acquainted with waveform manipulation, shift registers, and feedback nodes in LabVIEW.
- Get acquainted with the modularity of LabVIEW; more specifically, the usage of subVIs and LabVIEW LLBs.

1.2 Checkoff Points

2 Pre-Lab Section

2.1 Delay-Adder-Gain Block Diagrams

We have learned in lab 05 that discrete-time filters can be represented using **linear, constant-coefficient difference equations**, also known as **LCCDEs**. Given an LCCDE, we can construct a visual representation of a discrete-time filter through a **delay-adder-gain block diagram** that, as the name implies, employs delay elements, adder elements, and gain elements. We have achieved prior experience of this in lab 05 for simple LCCDEs. However, as we move onto systems with more complicated LCCDEs, we will now explore a more general method to draw delay-adder-gain block diagrams.

Consider the generalized LCCDE

$$y(n) = \sum_{k=0}^{M} \alpha_k x(n-k) + \sum_{l=1}^{N} \beta_l y(n-l), \tag{1}$$

where α_k and β_l represent scalar gains for the corresponding delayed signals. Note that the gain on the output signal is unity; it is possible for the $y(n)$ term to be preceded by a scalar nonzero gain, say β_0, but in that case, the entire LCCDE can be divided by β_0 to obtain an equivalent LCCDE.

One possible delay-adder-gain block diagram for this generalized LCCDE is shown in Figure 1. However, it assumes the existence of delay elements that can delay a signal by multiple samples. A more efficient delay-adder-gain block diagram is shown in Figure 2, born of the observation that delaying a signal by $k (> 1)$ samples is equivalent to delaying a signal by one sample, and then delaying *that* signal by one more sample, and so on. The newer delay-adder-gain block diagram assumes that we have, at our disposal, only delay elements that can delay a signal by one sample. This version has the added advantage of being easily implemented in LabVIEW: instead of using the specialized `Y[i] = X[i-n] PtByPt` block, we can use regular shift registers to implement the newer delay-adder-gain block diagram; more specifically, we can use shift registers with cascading delays, as was seen in lab 04, the lab on discrete-time convolution.

Figure 1 Delay-adder-gain block diagram for LCCDE 1.

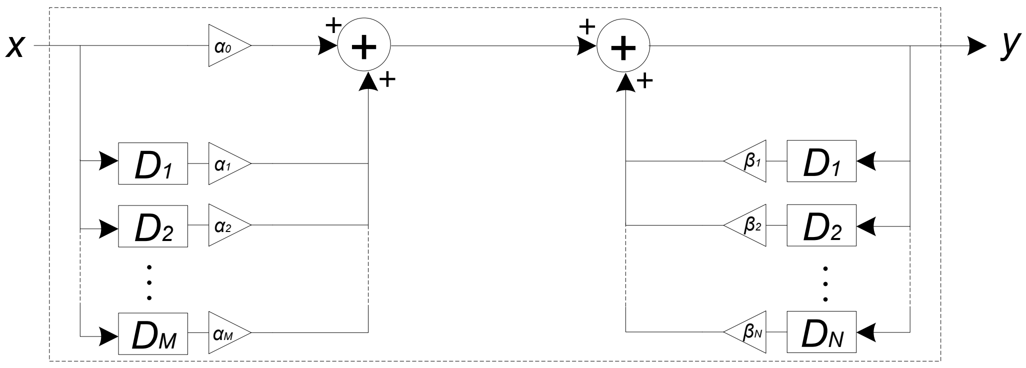
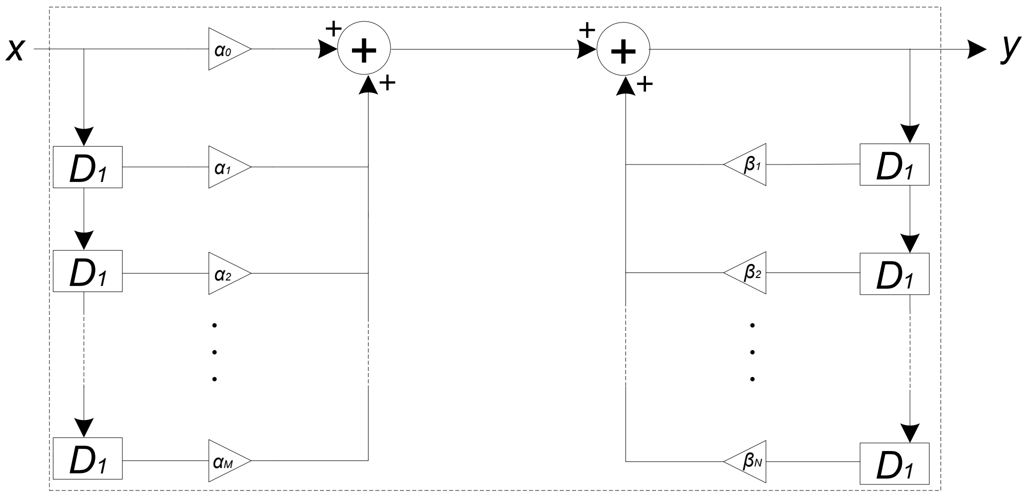


Figure 2 Another delay-adder-gain block diagram for LCCDE 1.



2.2 Geometric Interpretation of Frequency Responses

In this class, and in a lot of real-life applications, we concern ourselves with discrete-time filters whose frequency responses are rational in $e^{i\omega n}$. This means that both the numerator and the denominator of the frequency response (for, say, a system F) are polynomials in $e^{i\omega n}$. In fact, it can be shown that such a frequency response is of the form

$$F(\omega) = \frac{(e^{i\omega} - z_1)(e^{i\omega} - z_2)\cdots(e^{i\omega} - z_N)}{(e^{i\omega} - p_1)(e^{i\omega} - p_2)\cdots(e^{i\omega} - p_M)}. \tag{2}$$

We are interested in plotting such frequency responses; through these plots, we can understand, among other things, which frequencies are favored, which are attenuated, and which are amplified, allowing us to tweak necessary parameters to get the filter that we need. Unfortunately, since frequency responses are complex functions in general, we cannot plot these responses on paper, since we would require three axes (why?). As a result, we plot the magnitude and the phase of the frequency responses separately, and in doing so, we get a better idea as to how systems affect the magnitudes and the phases of signals with different frequencies.

Providing such plots, however, is not a trivial task. If we needed to generate a precise plot, we would look to a computer to do the plotting for us. For relatively simple frequency responses, however, we can sketch out the general shape of these plots, which is usually enough to afford us an idea of how our filter works.

2.2.1 Low-Pass Filter

For example, we consider the frequency response of a low-pass filter,

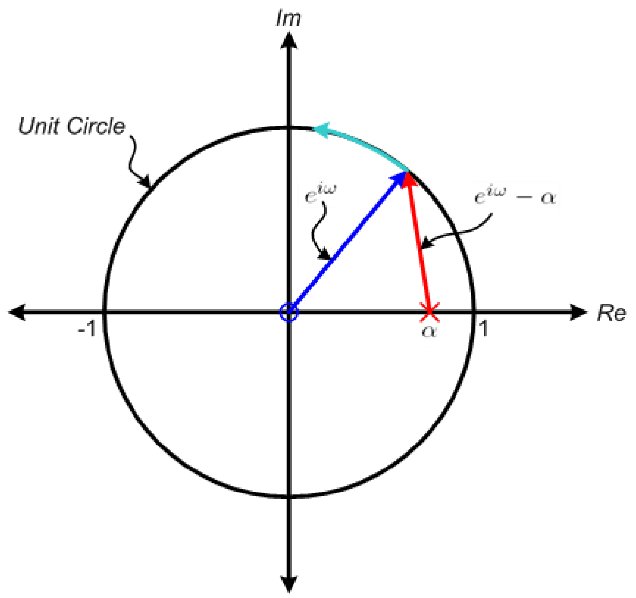
$$F_L(\omega) = \frac{e^{i\omega}}{e^{i\omega} - \alpha}, \quad 0 < \alpha < 1.$$

Rewriting the function slightly differently,

$$F_L(\omega) = \frac{e^{i\omega} - 0}{e^{i\omega} - \alpha}, \quad 0 < \alpha < 1. \tag{3}$$

Our main aim now is to draw vectors for each of the terms in the numerator and in the denominator. Now, comparing Equation 3 with Equation 2, we find that $z_1 = 0$ and that $p_1 = \alpha$. On the unit circle in the complex plane, we mark an `O` for every $z_i, 1 \leq i \leq N$ and we mark an `X` for every $p_i, 1 \leq i \leq M$[1]. We then draw the vector $e^{i\omega}$, whose tip rotates on the unit circle (why?). We draw vectors for the other terms, starting from each `X` and `O` and ending at the tip of the vector $e^{i\omega}$. Using laws of vector addition, we know that each of these vectors represents one term in Equation 3. A potential final result is shown in Figure 3.

Figure 3 Geometric Interpretation of a Low-Pass Filter.



Awesome! With this in hand, we can provide a qualitative sketch of the magnitude and phase responses of a low-pass filter. We know, for instance, that the magnitude of the frequency response is given by

$$|F_L(\omega)| = \left| \frac{e^{i\omega} - 0}{e^{i\omega} - \alpha} \right| = \frac{|e^{i\omega} - 0|}{|e^{i\omega} - \alpha|} = \frac{1}{|e^{i\omega} - \alpha|}.$$

Then, as the vector $e^{i\omega}$ moves about the unit circle, we need only determine how the magnitude (or length) of the vector $e^{i\omega} - \alpha$ changes. By looking at our diagram, we find that the length of the vector $e^{i\omega} - \alpha$

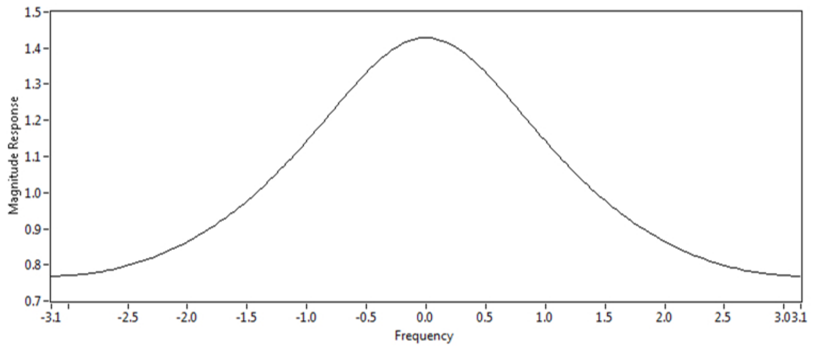
[1]In technical terms, the points we mark as `O` are called *zeros*, while the points we mark as `X` are called *poles*. However, in this class, we will not use these terms; for this class, we use the `X`s and `O`s merely to keep track of points, allowing us to remember which points are in the numerator, and which points are in the denominator.

is minimum when $\omega = 0$, and is maximum when $\omega = \pm\pi$. As a result, the magnitude of the frequency response is maximum when $\omega = 0$, and is minimum when $\omega = \pm\pi$. A quick evaluation tells us that

$$|F_L(\omega)||_{\omega=0} = \frac{1}{1-\alpha}, \qquad |F_L(\omega)||_{\omega=\pm\pi} = \frac{1}{1+\alpha}.$$

We collate all of this information and make a rough sketch of the magnitude response, similar to the computer-generated plot shown in Figure 4. Notice that the decrease in the plot, as ω goes from 0 to $\pm\pi$, is not a linear decrease. If we refer back to our diagram, we find that this is because the length of the $e^{i\omega} - \alpha$ vector does not change linearly either; initially, its length increases quickly, but later on, its length increases relatively slowly. We can use either calculus or a computer to figure out at precisely which point this change in concavity happens, but since we only care about the qualitative shape of the graph, we will not bother. However, we can now clearly see why this filter is a low-pass filter.

Figure 4 Magnitude of the Frequency Response of a Low-Pass Filter.



Correspondingly, the phase of the frequency response is given by

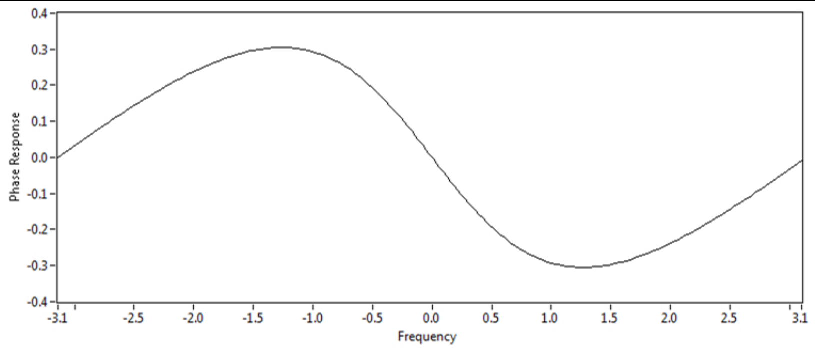
$$\angle F_L(\omega) = \angle\left(\frac{e^{i\omega} - 0}{e^{i\omega} - \alpha}\right) = \angle(e^{i\omega} - 0) - \angle(e^{i\omega} - \alpha).$$

This implies that we should look at the difference between the angles that the vectors $e^{i\omega}$ and $e^{i\omega} - \alpha$ make, respectively, with the real axis. We find that, as ω goes from 0 to π, the angle made by $e^{i\omega} - \alpha$ is always greater than the angle made by $e^{i\omega}$. Thus, the difference is negative, as ω goes from 0 to π, and by symmetry, the difference is positive, as ω moves from 0 to $-\pi$. A quick evaluation tells us that

$$\angle F_L(\omega)|_{\omega=0} = 0, \qquad \angle F_L(\omega)|_{\omega=\pm\pi} = 0.$$

Again, as ω goes from 0 to π, we notice that the angle made by the vector $e^{i\omega}$ changes at a steady rate (how much?), but the change in the angle made by the vector $e^{i\omega} - \alpha$ is really fast initially, but slows down as ω approaches π. As a result, the phase of the frequency response starts off at 0 and decreases fast, but it eventually evens out and starts to increase as it goes back to 0. A computer-generated plot for the phase of the frequency response is shown in Figure 5.

Figure 5 Phase of the Frequency Response of a Low-Pass Filter.



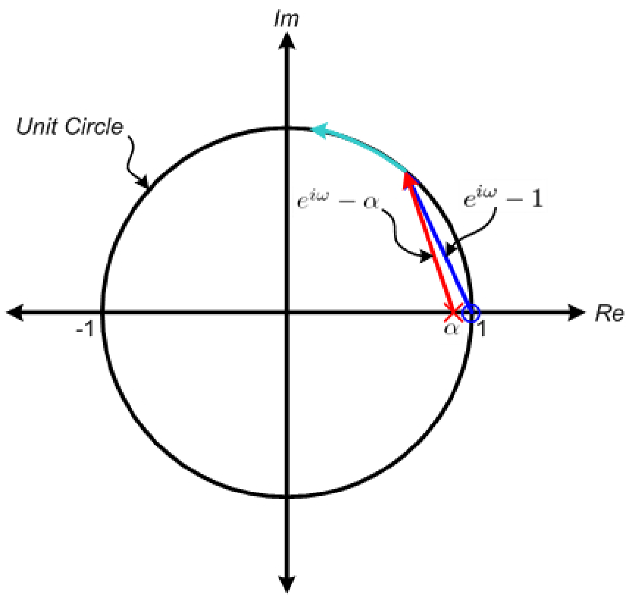
2.2.2 Notch Filter

As another example more relevant to this lab, we consider a different kind of filter, known as a *notch filter*. The motivation behind a notch filter is to attenuate several singular frequencies while preserving the rest. The frequency response of one such filter is given by

$$F_N(\omega) = \frac{e^{i\omega} - 1}{e^{i\omega} - \alpha},$$

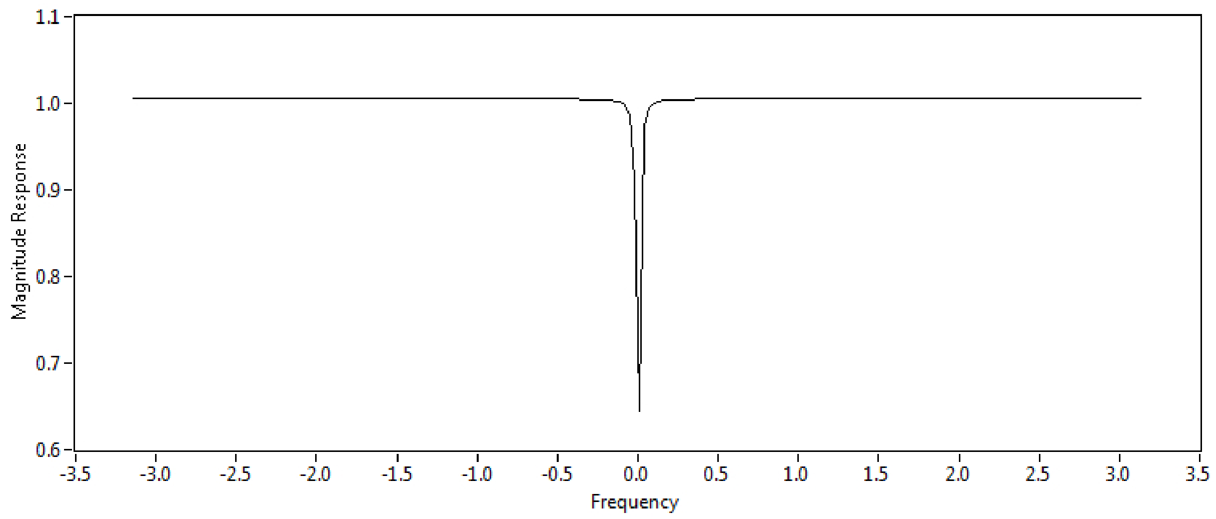
where α is really close to, but not equal to, 1. Following the algorithm presented in section 2.2, we draw the relevant geometric interpretation of the frequency response, as shown in Figure 6.

Figure 6 Geometric Interpretation of a Notch Filter, notching out the Zero Frequency.



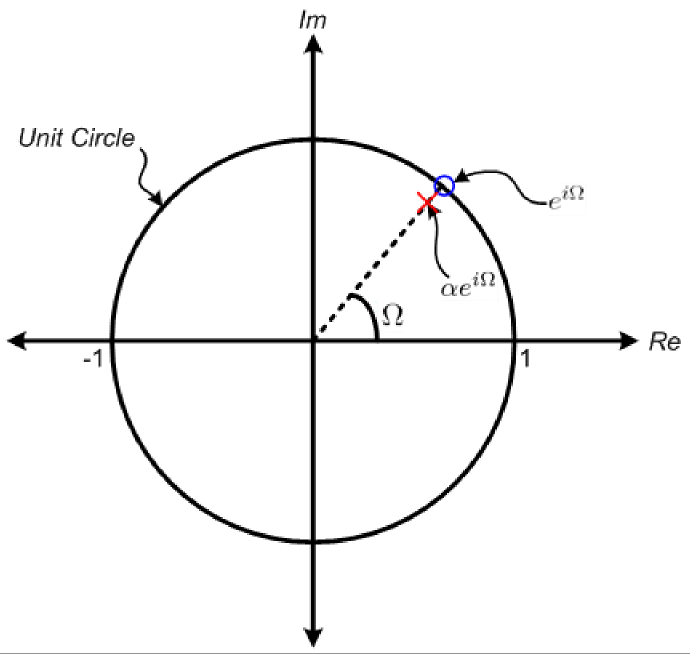
We notice that, for angles beyond a certain neighborhood around 0, the vectors $e^{i\omega} - 1$ and $e^{i\omega} - \alpha$ are approximately the same length; this approximation holds better as α gets closer to 1. However, in that small neighborhood around 0, the vector $e^{i\omega} - 1$ is of negligible length; in fact, at $\omega = 0$, the vector $e^{i\omega} - 1$ is precisely zero. As a result, the magnitude of the frequency response $F_N(\omega)$ is approximately 1 for all angles except around zero, where the magnitude is negligible. This filter is said to *notch out* the zero frequency. A computer-generated plot for the magnitude of the frequency response is shown in Figure 7.

Figure 7 Magnitude of the Frequency Response of a Notch Filter.



What would you have to do if you wanted to notch out another frequency – say, Ω – instead of the zero frequency? Where would you place the `X` and the `O`? Extrapolating from the case for zero frequency, we deduce that the `O` should be placed on the unit circle and at angle Ω. Thus, the `O` should be placed on the complex number $e^{i\Omega}$. Again, the `X` should be placed slightly away, but not too far away, from the `O`, and should also be at angle Ω. Thus, the `X` should be placed on the complex number $\alpha e^{i\Omega}$, where $|\alpha| < 1$, but α is very close to 1. The relevant geometric interpretation is shown in Figure 8.

Figure 8 Geometric Interpretation of a Notch Filter, notching out the Frequency Ω.



What would the frequency response be of such a filter? Recall, from section 2.2.1, that if the frequency response is of the form

$$F(\omega) = \frac{(e^{i\omega} - z_1)(e^{i\omega} - z_2)\cdots(e^{i\omega} - z_N)}{(e^{i\omega} - p_1)(e^{i\omega} - p_2)\cdots(e^{i\omega} - p_M)},$$

we mark an O for every $z_i, 1 \leq i \leq N$ and we mark an X for every $p_i, 1 \leq i \leq M$.

2.3 So You Think You Can Design a Notch Filter

For all of the in-lab sections, we will be concerned with the following problem: we have a signal $x(n)$, which in this case is a pure ECG signal. However, the signal that we obtain, $x'(n)$, is *not* a pure ECG signal; it is an ECG signal corrupted with a noise signal, $\eta(n)$. In other words,

$$x'(n) = x(n) + \eta(n),$$

and we are interested in filtering out $\eta(n)$.

For the purposes of this problem, we consider $\eta(n)$ to be the function $A\cos(\Omega n)$ ($A \in \mathbb{R}$), for several reasons: this noise signal simulates the actual problem well, and is a real-valued signal. Now, we know that the cosine function contains two frequencies – $\pm\Omega$ – and in order to remove the error signal from $x'(n)$, we need to notch out these frequencies. Thus, a notch filter will satisfy the requirements of this problem. Let us flesh out this notch filter, which we will call N.

1. We know, from section 2.2.2, how to create a filter that can notch out one frequency. Then, in order to notch out *two* frequencies, we should consider each frequency separately and notch it out separately. With this in mind, generate the necessary frequency response, $N(\omega)$, for the filter N. It should have the form

$$N(\omega) = \frac{(e^{i\omega} - z_1)(e^{i\omega} - z_2)}{(e^{i\omega} - p_1)(e^{i\omega} - p_2)}$$

   for some complex values of z_1, z_2, p_1, p_2.

2. Once we have the frequency response that we require, we need to determine the corresponding LCCDE, so that we can implement the filter in the time domain. We first consider the general case: determine the frequency response of a system F, which is specified by the general LCCDE

$$y(n) = \sum_{k=0}^{M} \alpha_k x(n-k) + \sum_{l=1}^{N} \beta_l y(n-l).$$

3. Based on your responses to step 1 and step 2, determine the LCCDE for the notch filter N. A few tips:

   (a) You may find it useful to bring $N(\omega)$ to the following form:

$$N(\omega) = \frac{1 + D_1 e^{-i\omega} + D_2 e^{-2i\omega}}{1 + D_3 e^{-i\omega} + D_4 e^{-2i\omega}},$$

   for some complex constants D_1, D_2, D_3, D_4.

   (b) Try and simplify your LCCDE as much as possible, so that the coefficients of each term are real; use any identities that you may know.

   (c) The final LCCDE should have the form

$$y(n) = [x(n) + C_1\cos(\Omega)\,x(n-1) + C_2 x(n-2)] + [C_3\cos(\Omega)\,y(n-1) + C_4 y(n-2)], \tag{4}$$

   where C_1, C_2, C_3, C_4 are constants that may, or may not, depend on α.

4. Draw the delay-adder-gain block diagram for your notch filter, based on the LCCDE 4 that you created in step 3.

Woah. What did you just do? You have not just designed a notch filter to notch out a particular signal, but you have also generated its LCCDE! Keep this LCCDE safe; the rest of the lab will be devoted to testing whether or not your notch filter does its job correctly.

2.4 Submission Rules

1. Submit your files *no later than* 10 minutes after the beginning of your next lab session.
2. Late submissions will *not* be accepted, except under unusual circumstances.
3. If the pre-lab exercises are not performed, you will get an immediate zero for the entire lab.
4. These exercises should be done *individually*.
5. Keep your work safe for further usage in the in-lab sections.

2.5 Submission Instructions

1. Log on to bSpace and click on the `Assignments` tab.
2. Locate the assignment for `Lab 6 Pre-Lab` corresponding to your section.
3. Answer the questions presented in section 2.3, and show your work. Templates for this assignment are available, in DOC and TEX formats, as part of the lab 6 resources on bSpace, but you need not use them.

3 Acknowledgments

Special thanks go out to the teaching assistants (TAs) of the Spring 2009 semester (Vinay Raj Hampapur, Miklos Christine, Sarah Wodin-Schwartz), of the Fall 2009 semester (David Carlton, Judy Hoffman, Mark Landry, Feng Pan, Changho Suh), and of the Spring 2010 semester (Xuan Fan, Brian Lambson, Kelvin So) for providing suggestions, ideas, and fixes to this lab guide.

References