DSP Laboratory: IIR Filter Design via the Bilinear Transformation*

Erik Luther

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

Abstract

The purpose of this lab is to examine IIR filter design/implementation using LabVIEW DSP. Particular attention is drawn to the comparison between theoretical filter characteristics and actual filter performance. In this lab, it is assumed that the student is already familiar with the basic operation of LabVIEW DSP.

1 IIR Filter Design via the Bilinear Transformation

The purpose of this lab is to examine IIR filter design/implementation using LabVIEW DSP. Particular attention is drawn to the comparison between theoretical filter characteristics and actual filter performance. In this lab, it is assumed that the student is already familiar with the basic operation of LabVIEW DSP.

1.1 Create a VI for Plotting Time and Frequency Domain Data

1. Open Surround Mixer, a program which controls the input/output operation of the sound card. Make sure that the Wave/mp3 and Line-in volumes are set to approximately 75%. Make sure that all source buttons are muted for now. Keep the Surround Mixer open throughout the lab. You will be selecting various inputs (Wave and Line-in) throughout the lab.
2. Start LabVIEW Embedded Edition (Start»All Programs»LabVIEW Embedded Edition) and open a blank VI (Virtual Instrument).

*Version 1.1: Sep 20, 2005 6:20 pm -0500
†http://creativecommons.org/licenses/by/2.0/

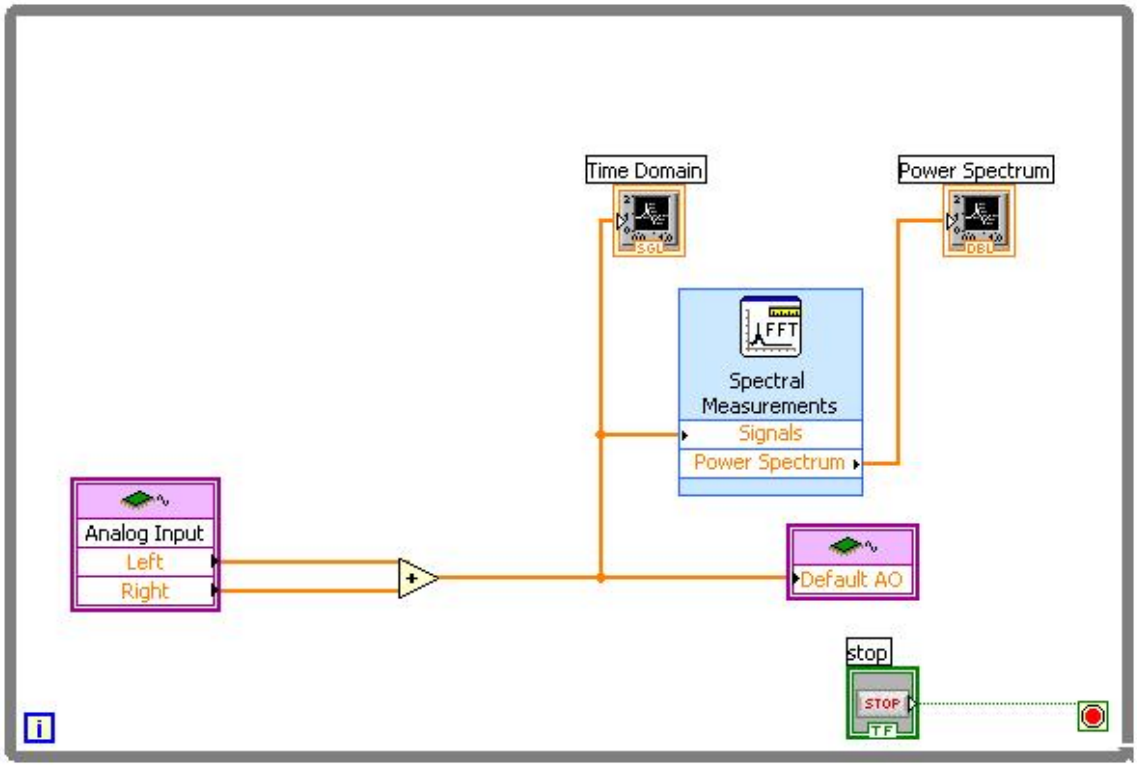

Figure 1: Pass-Through VI

3. We will build the VI whose Block Diagram is shown in Figure
   - Switch execution target to your hardware by going to Operate»Switch Execution Target in LabVIEW Embedded.
   - Go to the Block Diagram of the blank VI, navigate the function palette to the Analog Input, click and drop an analog input element on the Block Diagram. It is located under Functions»Elemental IO»Analog Input. See Figure 2.

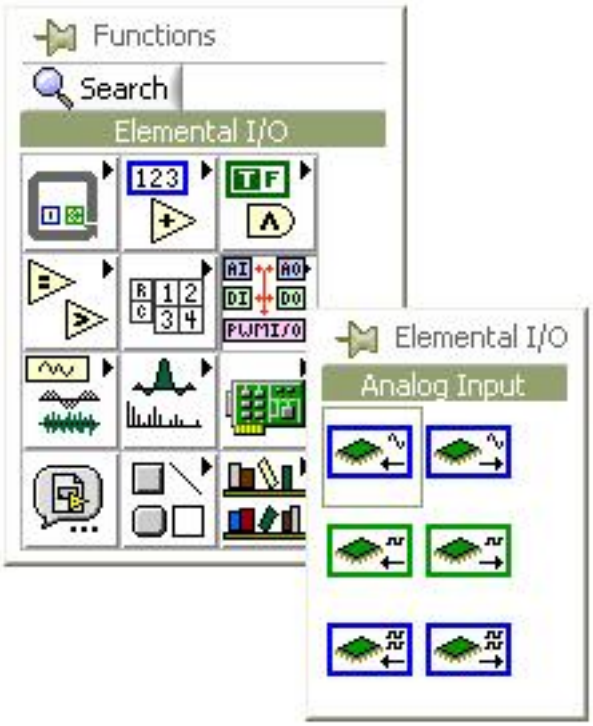

Figure 2: Elemental I/O sub palette

- Double-click on the Analog Input elemental I/O for configuration. In the configuration window, select the "General" tab. Click on the drop down under resource and select 2 channel Multiple Samples. On the Configuration tab, set the Sampling Rate to 16000 Hz and the Framesize to 256. Click "OK" to close the window.
- Drop an Add function located on the Numeric sub palette of the Functions palette: Functions»Numeric»Add.
- Drop an Analog Output elemental I/O. Functions»Elemental I/O»Analog Output. Double click on the Analog Output you just dropped on the Block Diagram and set the Sample Rate to 16000 Hz on the Configuration tab. Click "OK" to close.
- Drop a "Spectral Measurement" Express VI on the block diagram. Functions»Signal Processing»Frequency Domain»Spectral Measurements. This VI will allow us to build the power spectrum of the sound signal. Double click on the Spectral Measurements Express VI to open its configuration window. In the window, select Power Spectrum for Spectral Measurements and dB for results (See Figure)
- Click "OK" to validate the configuration.

Figure 3: Spectral Measurements Configuration

- Now let's switch to the Front Panel (Windows»Show Front Panel) and drop two Waveform Graphs. The Waveform Graph is located on the Graph Sub palette: Controls»Graphs»Waveform Graph. Name one "Time Domain" and the other "Power Spectrum." Do not add the stop button yet.

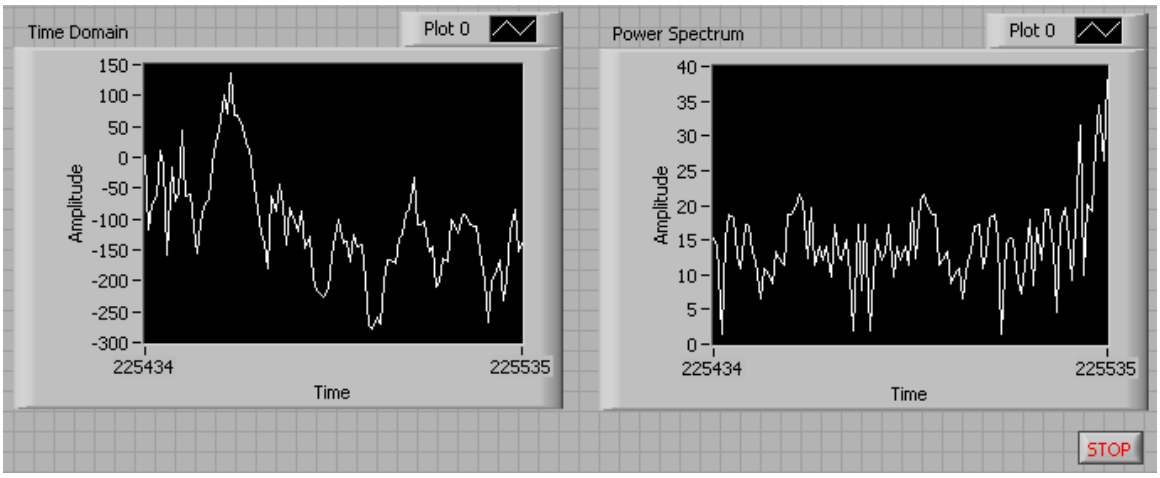

Figure 4: VI Front Panel

- Let's switch back to the Block Diagram and wire everything already on the diagram as shown in Figure 1.
- Select the while loop from Functions»Structures»While Loop, click and drag to enclose everything on the Block Diagram within the loop. You Block Diagram should now be similar to the one in Figure 1 and the Front Panel similar to the one in Figure 4.

  NOTE: This setup is similar to that found in the "DSP Laboratory: Time and Frequency Displays" Module except that the Left and Right channels are being combined into a single channel.

4. In Surround Mixer activate the Wave/mp3 source and put a CD into the computer's CD-ROM drive. Run the Windows Media Player program in the EE 453 folder to start the CD. Finally, press the Run icon and you should hear your music through the headphones. If there is distortion/clipping (due to the volume being too high), adjust the CD volume on the Surround Mixer to lower the volume until the clipping stops. Now adjust the two displays to make one a time-domain display and the other a frequency-domain display. Remember how to set/unset the automatic scaling on the displays. It is usually best to use Auto scaling on the frequency-domain display but not the time-domain display.
5. Once you have verified that everything is working properly, stop the CD, then stop the LabVIEW program (but don't exit LabVIEW).

6. 11. Save the VI by selecting File»Save As. Make sure to save the file in your personal directory – Desktop\ee 453\<folder name>. You can give the file any name you want (such as lab2setup).

1.2 IIR Filter Design via the Bilinear Transformation

LabVIEW DSP allows the user to implement standard IIR digital filters designed using a filter design supplemental program, called Digital Filter Design Toolkit. The DFD toolkit is an add-on module to LabVIEW (not LabVIEW Embedded). This module allows the user to design complex filters, save them and implement them using LabVIEW DSP.

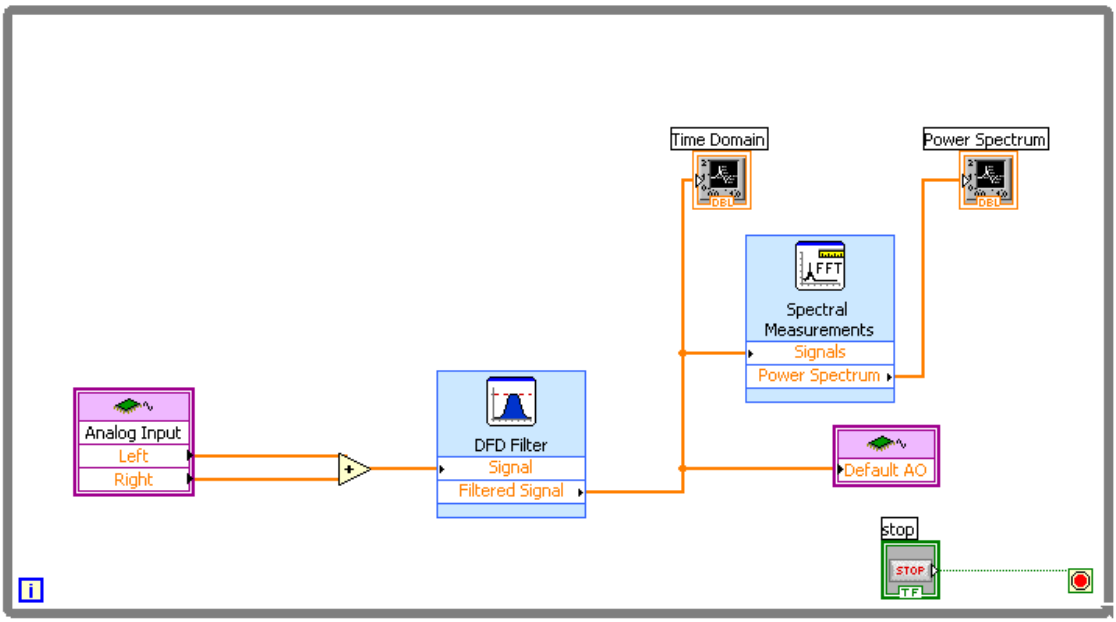

Figure 5: New Block Diagram with DFD Filter

1. Go to the Functions palette and add the DFD Filter Express VI to your existing Block Diagram: Functions»Signal Processing»Filters»DFD Filter
2. Wire the Express VI as shown in Figure 6. The output of the Add function is the input "Signal" of the Express VI and the "Filtered Signal" goes into the Analog Output. Set the function generator's signal to a frequency of 500 Hz.
3. Follow the steps below to implement a lowpass filter that will be implemented on the DSP. Launch LabVIEW by going to Start»All Programs»National Instruments»LabVIEW 7.1»LabVIEW. Let's try to understand what is happening. We first launched LabVIEW Embedded Edition which allowed us to target the DSP hardware and run our program. Now, we are launching LabVIEW, not LabVIEW Embedded Edition. The program you built earlier should still be open. We will use the Digital Filter Design Toolkit available in the LabVIEW environment to design the filter.
4. Create a blank VI (New»Blank VI).

5. Switch to the Block Diagram and drop a DFD Classical Filter Design on the Block Diagram: It is in Functions»All Functions»Digital Filter Design»Filter Design»DFD Classical Filter Design.
6. In the DFD Classical Filter Design Express VI, enter the following settings:
   - Filter Type: Lowpass
   - Sampling Frequency: 16000 Hz
   - Passband Edge Frequency: 1000 Hz
   - Passband ripple: 3 dB (standard for Butterworth filters)
   - Stopband Edge Frequency: 1500 Hz
   - Minimum stopband attenuation 20 dB
   - Design Method: Butterworth

   NOTE: Leave the other settings to their default values.

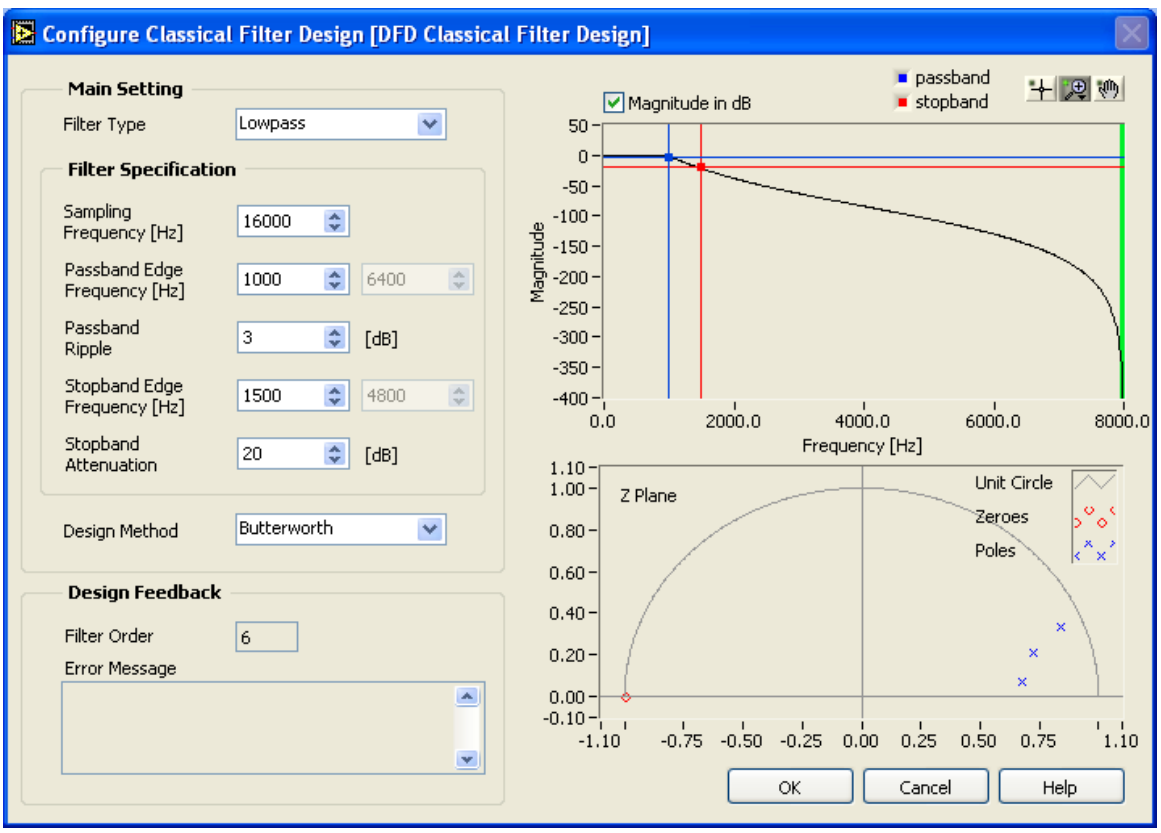

Figure 6: DFD Classical Filter Design Configuration Window

7. Check the graph to the left to make sure the filter response appears as expected and click "OK" to set the configurations.
8. Some filter parameters are described below:
   - Sampling Frequency – self-explanatory. Make sure to set it to the sampling frequency that you plan on using!
   - Center Frequency – (not used for lowpass filters). Set this to the middle frequency of the passband for Bandpass/Highpass filters or the middle frequency of the stopband for Bandstop filters.

- Bandwidth – Set this to the frequency width of the passband (stopband for Bandstop filters). Recall that Lowpass filters have a lower passband frequency of 0 Hz and that Highpass filters has an upper passband frequency of Fs/2.
- Stopband Attenuation – Set this to the desired attenuation in dB at the edge of the stopband. For Bandstop filters, this parameter must be negative (because the stopband is really the passband). For all other filter types, this parameter must be positive.
- Passband Ripple – Set this to the desired attenuation (in dB) at the edge of the Passband. For Butterworth filters, the standard (but not required) value to use is 3 dB.
- Filter Order N – Set this to the desired filter order. However, after you press the Apply button, the program calculates which filter order you need to meet the given specs. For most applications, then, you can put any number here and then change it to the necessary value when instructed. There may be some situations, however, when you want to force the filter order to be some set value. In this case, the desired specifications may not be met.

  NOTE: Regarding calculation of filter order N: This program uses the same design equations that we use in EE 453. For IIR filters, the program automatically does prewarping.

9. Build the following Block Diagram to save the filter settings: Drop a "DFD Save to File" VI on the Block Diagram. This VI is located at Functions»All Functions»Digital Filter Design»Utilities»DFD Save to File. Wire the Block Diagram as shown in Figure 6.

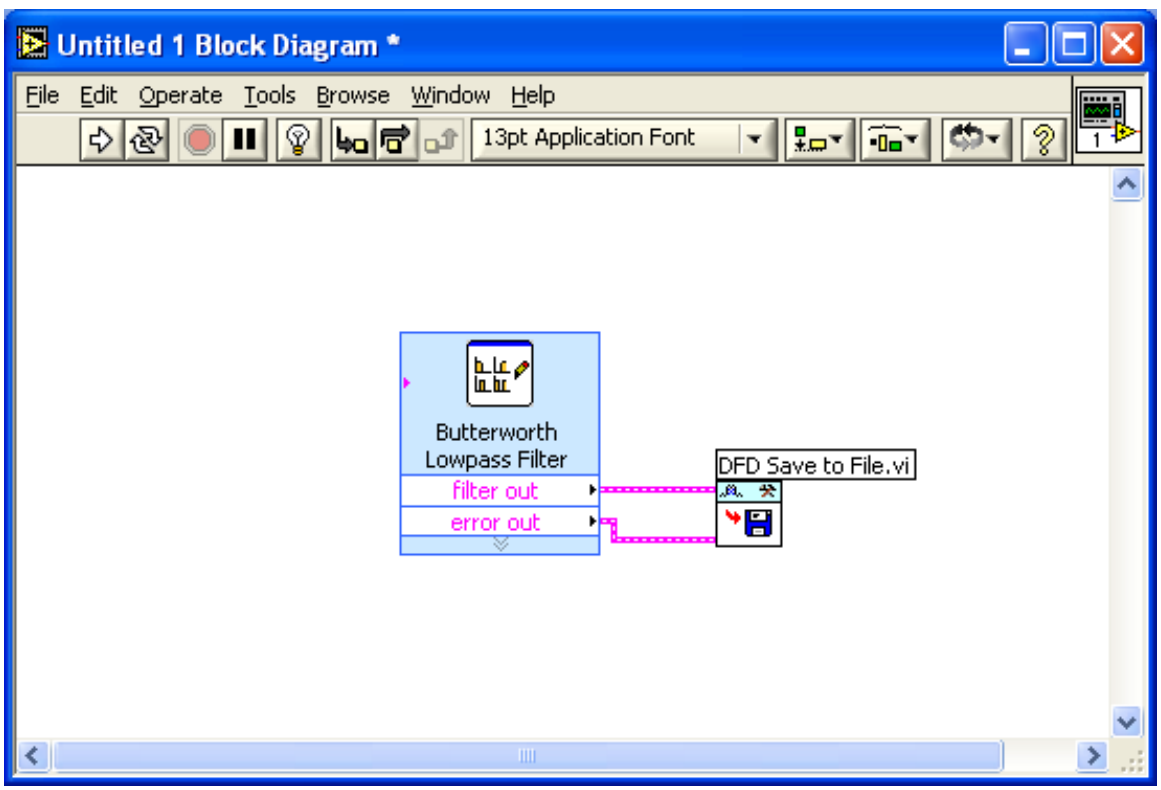

Figure 7: Filter Design

10. Run the filter design VI you just built. When prompted to save your design, navigate to your folder and save your filter under any chosen name. Save and close this VI.

1.2.1 Answer These Questions

Exercise 1
Calculate, by hand, the predicted filter order needed to meet the specifications of the lowpass filter. Show your work below.

Exercise 2
Does your predicted filter order agree with the filter order suggested by LabVIEW DSP? YES / NO

1.3

1. Let's now go back to the original DSP program and add the newly designed filter. Switch to the Block Diagram and Double-click on the DFD Filter Express VI you added on the Block Diagram in step 1. In "Path" section of the "Configure Filter" prompt, click the Browse button and navigate to the filter design you saved earlier, click "OK" to accept. Run the program.
2. Because the signal frequency is significantly lower than the passband cutoff frequency, the signal should pass through the filter with very little attenuation. Change the frequency plot to manual scale. Sweep the input signal frequency through the range 500 Hz. – 3000 Hz. and note the resulting time-domain and frequency-domain plots.
3. Set the input sine wave to the following frequencies: 1000 Hz, 1200 Hz, 1500 Hz, 2000 Hz, and 3000 Hz. For each frequency input, observe the corresponding filter output (both displays as well as the actual sound) and measure/record the exact amplitude of the output signal for each frequency. The easiest way to measure the amplitudes is to stop the simulation and then place your mouse pointer at the appropriate position on the graph. Note: You will need to change the scale of the graphs to read accurately. Call your lab assistant for help on how to change the scale.

1.3.1 Answer These Questions

Exercise 3
Record the amplitude and corresponding attenuation (in dB) of the filtered sinusoid for each of the frequencies below.

The equation for attenuation is: `attenuation = -20log(output amplitude/(maximum amplitude.`

Frequency	Output Amplitude	Attenuation (in dB)
500 Hz		0 dB, since log(1) = 0
1000 Hz		
1200 Hz		
1500 Hz		
2000 Hz		
3000 Hz		

Figure 8: Table for recording Problem 3 Results.

Exercise 4
Take special note of the attenuations at 1000 Hz and 1500 Hz. What do you expect these values to be theoretically? (show work) How do your actual values compare to the theoretical values?

1.4

- Put the input frequency to 800 Hz and run the program again. Without changing anything else, change the sampling frequency in the Analog Input and Analog Output elements to 8000 Hz and run the program. Observe what happens to the output signal. When done, set all sampling frequencies back to 16000 Hz.

1.4.1 Answer These Questions

Exercise 5
Explain what happens to the filter's performance when the sampling frequency is changed from Fs = 16000 Hz to 8000 Hz. Specifically, what happens to the 800 Hz signal when it is passed through the filter now? Explain what this implies about the importance of operating a digital filter using the same sampling frequency that you used to design it.

1.5

- Now deactivate the Line-In input in Surround Mixer and run the program again using your CD input instead. Listen to and observe the output of the filter (in both the time and frequency domain).

1.5.1 Answer These Questions

Exercise 6
Discuss the effect of the lowpass filter on the music signal.

1.6

1. To really see the effect of the filter more graphically, remove the Analog Input element and Add function from the Block Diagram and add a EMB Uniform White Noise Waveform node instead (Functions»Embedded Signal Generation»EMB Uniform White Noise Waveform.vi). This block generates white noise. White noise has energy spread equally across the frequency spectrum. For this reason, a white noise generator is often a good test of a filter's performance. Connect this block to the input of the IIR Filter. Double click on the Noise block and set the Amplitude = 10000, save and close it.
2. Run the program and observe the sound of the noise and take note of both its spectrum and time-domain shape. Now remove the lowpass filter block completely from the worksheet and connect the output of the Noise block directly to the Analog Output element. Re-run the program and observe the signal again. This is what unfiltered white noise looks/sounds like.

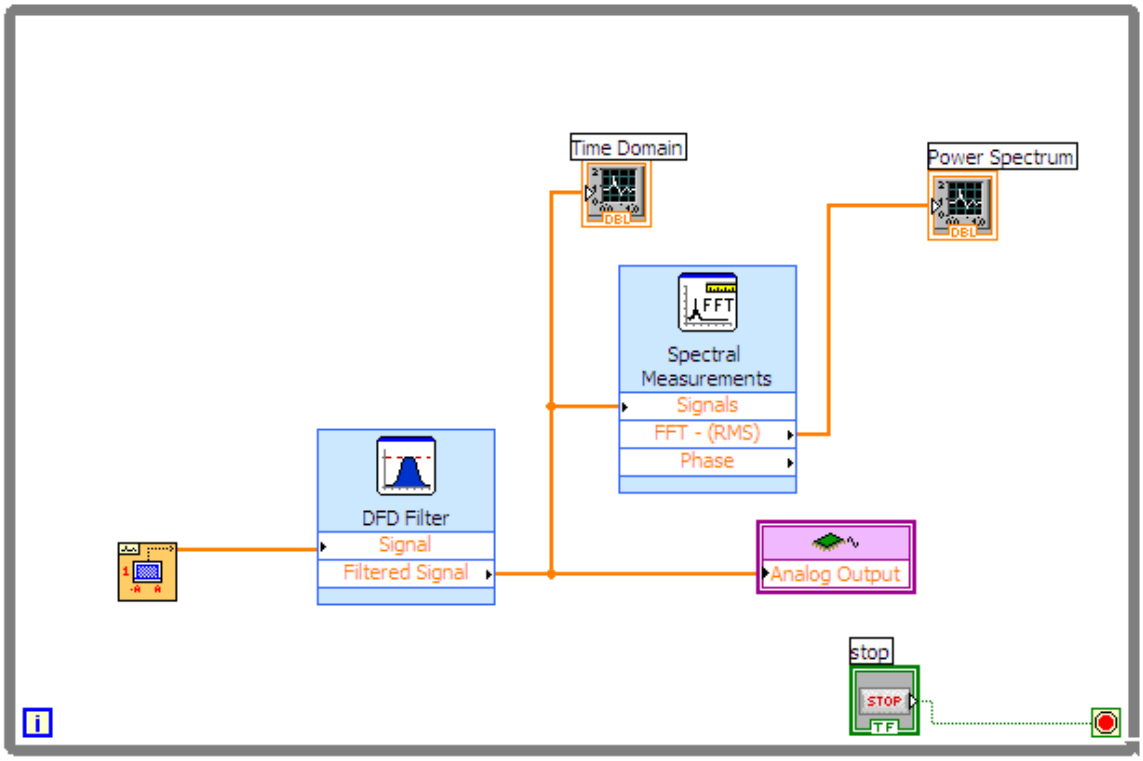

Figure 9: Program with Noise Generator

1.6.1 Answer These Questions

Exercise 7

Discuss how the shape of the white noise signal changes when it is passed through the lowpass filter. Also, describe how the sound changes.

1.7

- Now follow steps 4 to 7 to design a bandpass filter to meet the following specifications:
  - Type: Chebyshev I
  - FSTL: 1000 Hz
  - FPL: 1500 Hz
  - FPU: 3500 Hz
  - FSTU: 4000 Hz
  - Passband ripple: 1 dB
  - Minimum stopband attenuation: 40 dB
  - Sampling Frequency: 16000 Hz

- Save the filter to your folder (giving it a descriptive name like bpcheby1500-3500) and examine the various tabs in the filter design program.

1.7.1 Answer These Questions

Exercise 8
What is the IIR filter order needed to meet the bandpass filter specifications given? (You do not have to verify this value, just state what it is).

1. Re-add an IIR filter block back into your system and connect it to the system the same way you did the previous IIR block. Load your new filter coefficient file into the IIR filter block. Keeping the Noise Generator as your input signal, run the program again and note the difference between how the noise gets filtered by the bandpass and lowpass filters.
2. Remove the Noise block and re-add the Analog Input node and the Add Function. Connect these to the rest of the system as you did earlier and run the system using your CD input. Observe the output signal.
3. Add a second IIR filter block to your system in parallel with the first block. Load the original lowpass filter coefficient file into this block. Double-click on the Analog Output Element and select 2 channels multiple samples. Now send the output of one filter to one of the channels of the Analog Output element and the output of the other filter to the other channel. (Note: Only one of the filter outputs can be connected to the displays. It doesn't matter which one you have connected). Run the program using your CD input, and listen to the filtered output. You should hear a different signal in each ear.
4. As a final experiment, break the parallel filter connections and connect the 2 filter blocks in cascade rather than parallel by passing the output of one filter through the other filter and sending the output of the 2nd filter to the Analog Output element. Run the program using your CD input.

1.8

1.8.1 Answer These Questions

Exercise 9
When the lowpass and bandpass filters are cascaded together, what happens to the output signal? Why?

1.9

NOTE: Call your TA over to verify that you have completed this last part of the lab. If time permits, feel free to design a few other types of IIR filters and connecting them in various ways.