Lastname: ____________________

5Digit SS:____________________

LABORATORY
IIR Filters

LAB IIR: INTRODUCTION TO IIR FILTERS
(10 points)

In this lab, you will be introduced to the design of Infinite Impulse Response (IIR) filters. Filters are one of the most important elements in DSP and are typically used to isolate a specific frequency band of a signal. IIR filters are of particular interest because, with just a few coefficients, relatively sharp transition bands can be realized.

In the **pre-lab,** you will use "pencil-and-paper" to compute the results you expect later in your design implementation. In the **design part,** you will design a first-order IIR filter and a third order system direct form filter.

Lab Objectives

After completing this lab you should be able to

- Design and simulate an first order IIR filter
- Determine magnitude, phase and pole zero diagram of IIR filters
- Design a 3. order elliptic low pass filters
- Compare IIR and FIR design parameter

Pre-lab (3 points)

1. For a first order IIR filter with a transfer function $H(z)=b/(1+az^{-1})$, determine a and b such that the filter is a halfband filter (i.e., $|H(\omega=0)| = 1$ and $|H(\omega=\pi/2)| = 0.5$).

Feedback gain a = ____________.

Forward gain b = ____________.

Hint: The quadratic equation $x^2+px+q=0$ has the solution $x_{1,2}=-p/2+/-\mathrm{sqrt}((p/2)^2-q)$.

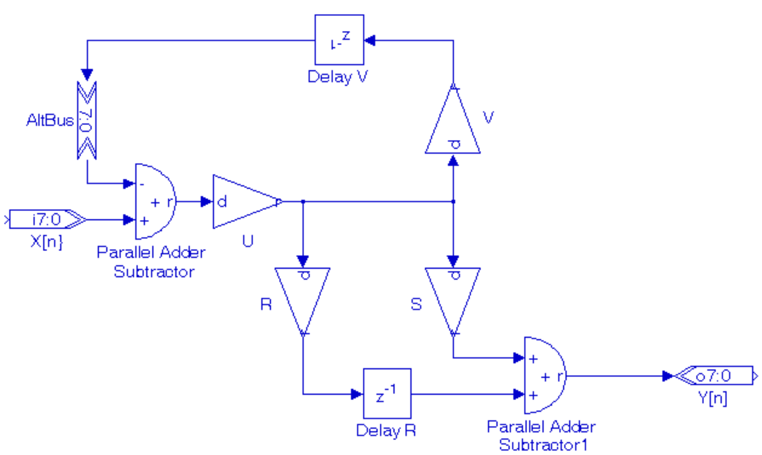


Figure 1

2. Determine the transfer function for the system in Figure 1.

$$H(z) = Y(z)/X(z) =$$

3. Determine zero(s) and pole(s) of the system in terms of the coefficients U,V,R, and S.

   Zero(s) at = _______________________ Pole(s) at = _______________

4. Find the values of the coefficients (U, V, R, S) so that the transfer function $H(z)$ from part 1 is realized

   U =

   V =

   R =

   S =

5. Compare the FIR and IIR filters regarding the following properties:

	FIR filter	**IIR filter**
Filter Length		
Filter Linearity		
Coefficient design method		
Pole locations		
Coefficient sensitivity to quantization		

Lastname: ________________

5Digit SS:________________

LABORATORY
IIR Filters

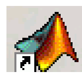
Simulink Design-lab (7 points)

Follow the directions below to implement a first and third-order IIR filter.

A. Getting Started

If you are in B114 or the digital logic lab:

1. On the desktop double click on **Engineering folder**.

2. Double click on the **MatLab** icon  to start **MatLab**. From the to top icon list in the **MatLab** window click on the **Simulink** icon  to start **Simulink.**

3. You should not save anything on the local hard disk. You will have to use a USB flash drive or your “mapped” home directory to save the files. Create a New Folder named **DSPwFPGAs** (if you have not so in a previous lab) on your mapped network drive.

B. Compiling an First-Order IIR Design

1. Download the file `iirorder1.mdl` and `showfft.m` from the class webpage and put the file in the **DSPwFPGAs** folder.

2. Click on the “Current Directory” selection icon and select your **DSPwFPGAs** folder as the current directory.

3. The files in the **DSPwFPGAs** can now be easily accessed with the "open file" button  on the **MatLab** toolbar**.** Double click on the `iirorder1.mdl` file and after a moment you should see the incomplete design:

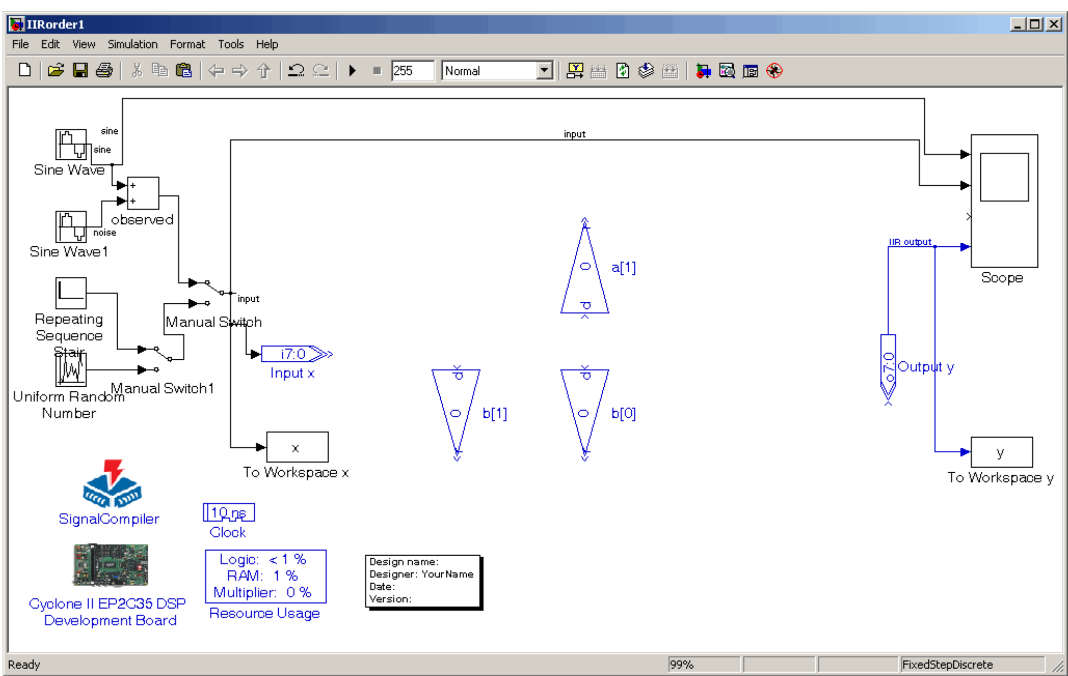


Figure 2: Incomplete First-Order IIR Filter.

4. Complete the design:
   a. Change the design coefficients to those you computed in part 1 of the pre-lab. This can be done by double-clicking in the Gain blocks and in the Parameter window changing the **Gain Value**. Use a 1.12 bit format and 0 for **Number of Pipeline Stages** for all coefficients.
   a. Copy the adder and delay elements from the Altera library according to Figure 1 from the pre-lab. Follow the steps described below to complete the design:

      - Click on the **Library Browser** icon and select the **Altera DSP Builder** directory.
      - Under the **Arithmetic** subdirectory, choose the **Parallel Adder Subtractor** and drag it into your workspace.
      - Double click on the **Parallel Adder Subtractor** to change its parameters. Deselect the **Enable Pipeline** option and for **Number of Inputs** choose 2 and in the **Add (+) Sub (-)** field write in -+ as shown below:

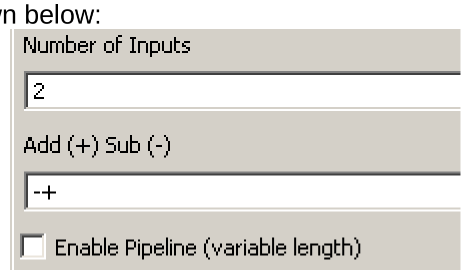


Figure 3

      - Connect the **Input** block to the + input of the **Parallel Adder Subtractor**.
      - Place a wire from the output of the **Parallel Adder Subtractor** to the input of **Gain** block a[1], then connect the input of **Gain** block b[0] and b[1] to this wire.
      - Under the **IO & Bus** subdirectory, choose the **Altbus** block and place it in your workspace. Rotate the block by choosing **Rotate Block** under the **Format** menu or doing *CTRL+R*. Connect the – input of the **Parallel Adder Subtractor** to the **Altbus**.
      - Add the **Delay** block from the **Storage** subdirectory. Flip the **Delay** block by either choosing **Flip Block** from the **Format** menu or doing *CTRL+I*. Connect the **Delay** input to the output of **Gain** block a[1] and the **Delay** output to the input of the **Altbus** block. Also connect the remaining output of the **Scope** to the input end of the **Altbus** block.
      - Follow the above steps to add another **Parallel Adder Subtractor** or simply cut and paste the one already existing on the workspace. Change the **Add (+) Sub (-)** parameter to + and deselect the **Enable Pipeline** option. Connect the output of **Gain** block b[0] to one of the + inputs of the **Parallel Adder Subtractor**.
      - Add a **Delay** block following the instructions above and connect the output of **Gain** block b[1] to the input of the **Delay**. The output of the **Delay** should be connected to the other + input of the **Parallel Adder Subtractor**.
      - Wire the output of the **Parallel Adder Subtractor** to the **Output y.**

5. At the MatLab prompt, use the predefined functions `freqz()` and `zplane()` to plot the frequency spectrum and pole/zero plot, respectively. First, you must define the numerator and denominator polynomial coefficients in vector form.

   - Recall the transfer function you found in the pre-lab, of form
     $H[z] = b[z]/a[z] = (b[0]+b[1]z^{-1}+...) / (a[0]+a[1]z^{-1}+...)$

   - **Use your coefficient values** to define the coefficient vectors. At the MatLab prompt, type the following:

     >> **b=[b[0], b[1], ...]; a=[a[0], a[1], ...];**

- Now, call the functions to generate the desired plots:

```
>> freqz(b,a)
>> zplane(b,a)
```

6. Complete the following diagrams:

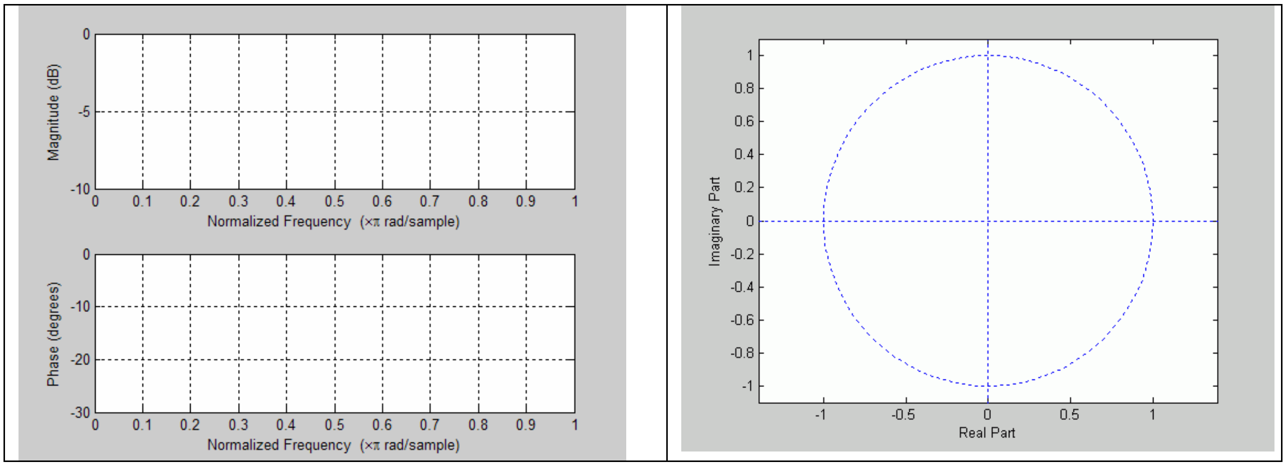


Figure 4: Fill-in Magnitude, Phase and Pole/Zero Plot for First-Order IIR Filter.

7. Simulate the design with the two sine input signals. In the MatLab prompt use `showfft(x)` and `showfft(y)` to display the spectra. Determine the 2nd sine (i.e., noise) component amplitude before and after filtering:

Before Filtering, $H(\omega_2)$ = ______

After Filtering $H(\omega_2)$ = ______

Note: Make sure the `showfft.m` file is in your current directory!

8. Determine the number of logic elements, DSP 9x9, and the minimum slack for the design:

Logic Cells = ________________

DSP 9x9 = ________________

Slack = ________________

C. Designing a 3rd Order Direct-Form Filter

1. Download `IIRorder3.mdl, setup_iir3.m, and showfft.m` from the class webpage and put the files in the **DSPwFPGAs** folder.

2. Open the file `setup_iir.m` with a text editor. There you will find the filter coefficient and the call to the predefined MatLab function `freqz()` and `zplane()` for the spectrum and pole/zero plot, respectively. In the MatLab prompt, type `setup_iir3` and complete the following diagrams:

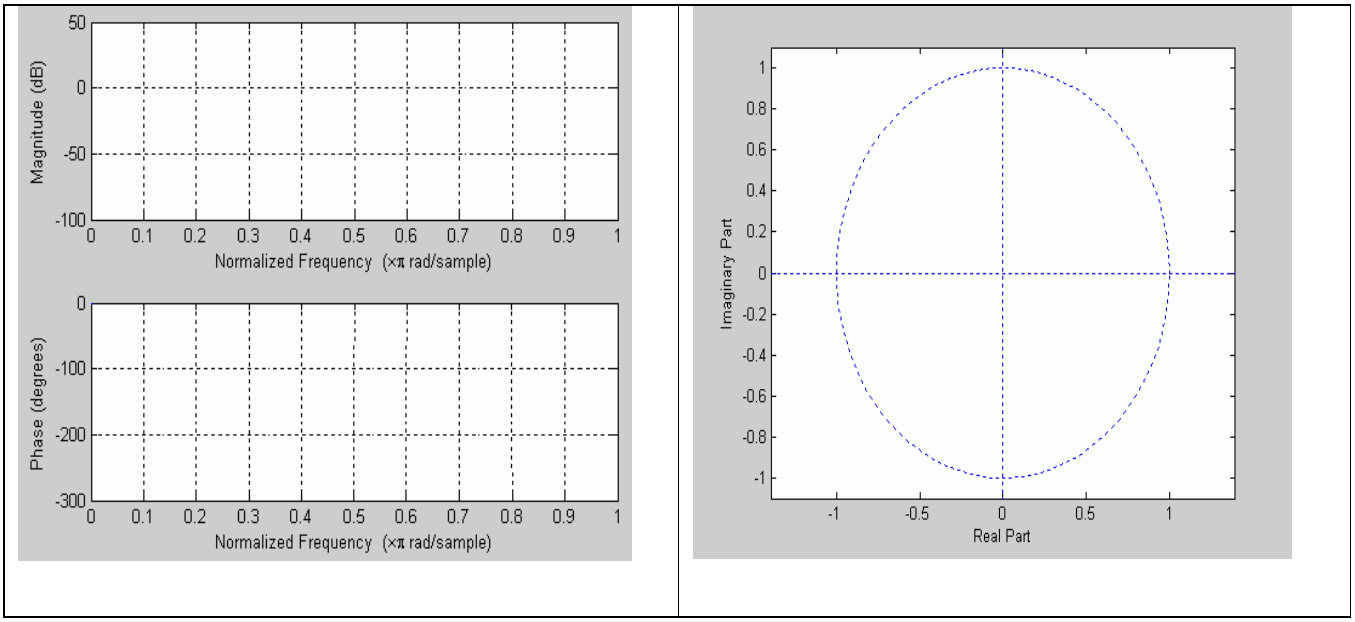


Figure 5: Fill-in Magnitude, Phase and Pole/Zero Plot for Third-Order IIR Filter.

3. Complete the design: add the delay and adder elements and change the values of the gains (all in 2.5 bit format) according to those in the `setup_iir3.m` file and the instructions from part B.

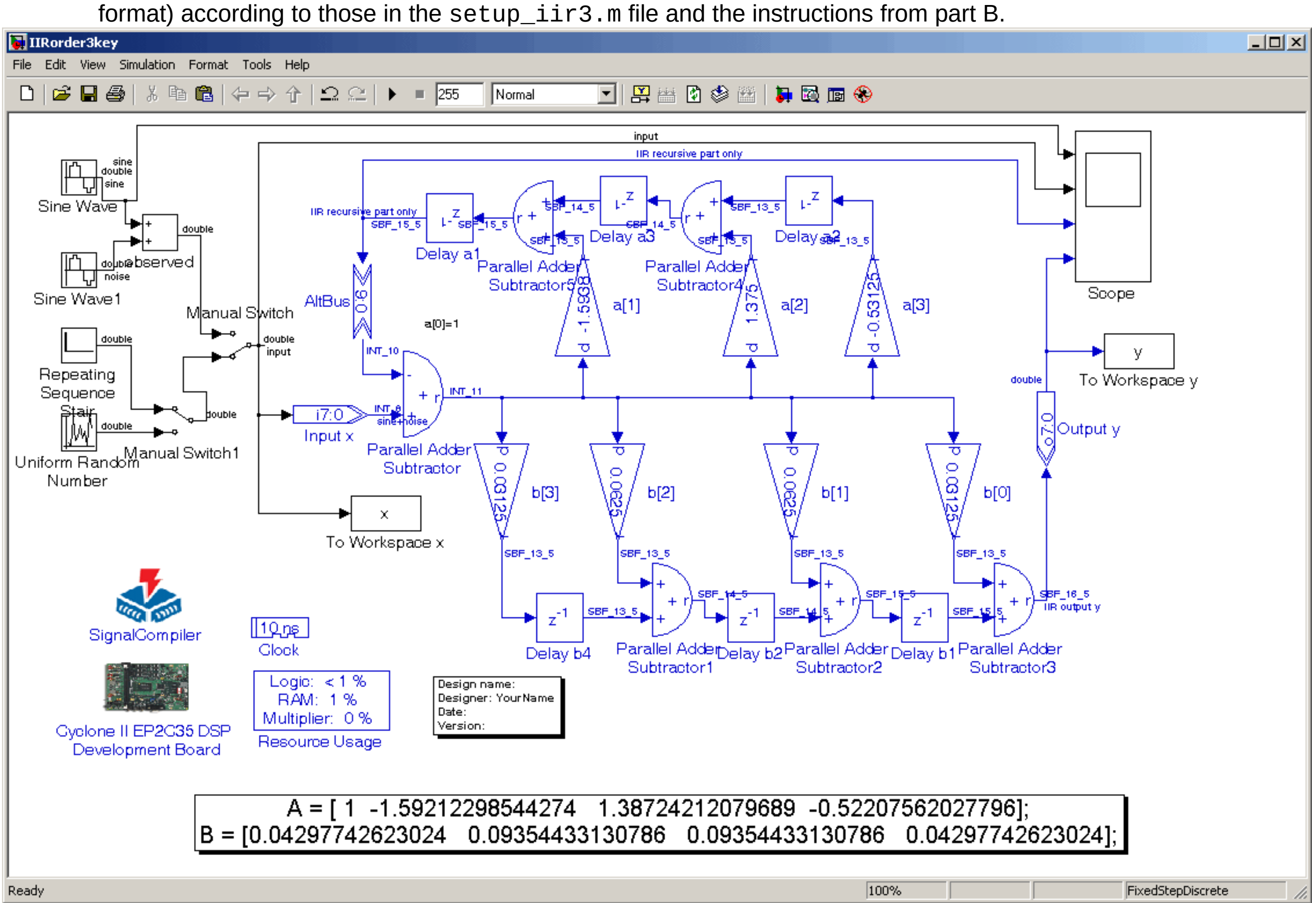


Figure 6: Third-Order IIR Filter.

4. Set the manual switches such that the impulse signal is the IIR filter input. Simulate the design using the **impulse** input signals and complete the spectra using use `showfft(x)` and `showfft(y)`:

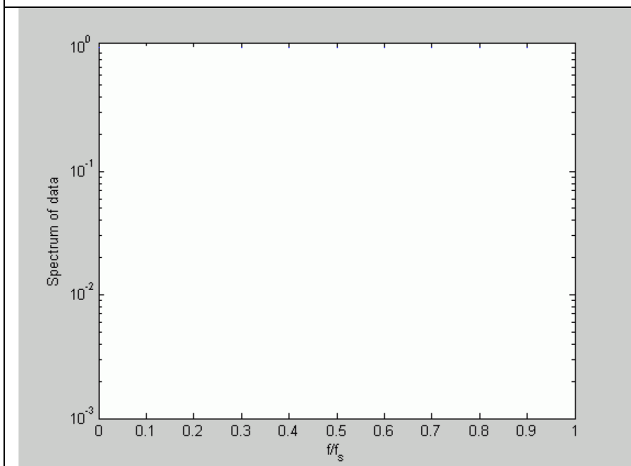
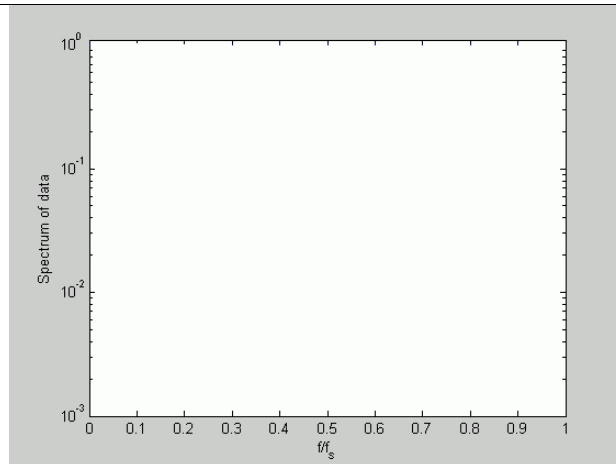
<table>
<tr><th>Input spectrum</th><th>Output spectrum</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Figure 7: Fill-in Third-Order IIR Impulse Response Spectra.

5. Set the manual switches such that the sine signals are the IIR filter input. Simulate the design using the sine input signals and complete the spectra:

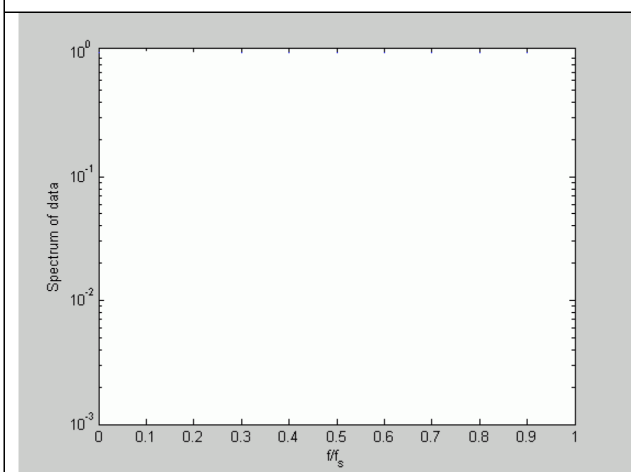
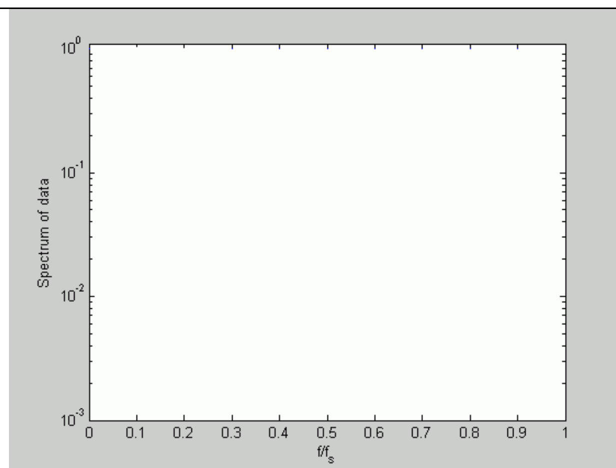
<table>
<tr><th>Input spectrum</th><th>Output spectrum</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Figure 8: Fill-in Third-Order IIR Response to Sine Waves Spectra.

6. Use the function `showfft(x)` and `showfft(y)` provided to “measure” the amplitude of the “noise” sine component before and after filtering. Compare the results to the first order system.

Before Filtering, $H(\omega_2)$ = ______

After Filtering $H(\omega_2)$ = ______

7. Compile the design using **Signal Compiler** and determine

Logic Cells = __________________

DSP 9x9 =___________________

Slack = __________________

from the report files or the **Resource Usage** block.

F. Deliverables:

1. Solve the problems of the pre-lab. (3 points).
2. Print the 2 MDF files and the 2 Simulink simulations for the two sine sum signals (7 points).

Make sure your name and SS is on all pages you turn in!