Faculty of Applied Science and Technology

Laboratory 5 DT Systems Analysis in the Frequency Domain

Student Name: Michael McCorkell Student Number: N01500049 Date: March 18th 2025

Note 1: This is an individual lab. Please bring your laptop and install required software to complete this lab.
Note 2: For your lab report, please take necessary screenshots and proper captions and include them in the corresponding parts of the lab. Demonstrate your work during the lab time. Submit the report file and the related .m files on the course website.

1. Learning outcome:

1.1 Use MATLSB Simulink to analyze DT LTI systems in the frequency domain.

1.2 Implement digital filters using MATLAB filter design tools.

2. Background

2.1 Simulink in MATLAB:

Simulink is a model-based MABLAB design tool. It creates virtual models to mimic the real-life systems and run simulations through the model. The benefit of Simulink and other simulation tools in the context of signals and systems is that it will put a clearer perspective on how systems behave through various signals. And you will be able to test your design and theory rapidly and iteratively through simulations.

The topics related to Simulink are vast and broad. MATLAB has offered a 3-hour introductory tutorial on Simulink. The course link is here. However, please keep in mind that this lab will only cover some introductory digital examples of how Simulink works in tandem with other MATLAB design and analysis tools. Completing the free MATLAB tutorial is not mandatory for the completion of the lab.

Simulink can be downloaded through the following steps: home ->Add-Ons ( )-> Get Add-Ons, search for the **Simulink** and download.

Accompanying Simulink, we also need to install the DSP System Toolbox through the following steps: home ->Add-Ons ( )-> Get Add-Ons, search for the **DSP System Toolbox** and download.

2.2 Digital systems analysis through simulation:

Figure 1. Simulink Download

So far, we have implemented z-transform and computed transfer functions through lectures and labs. In some cases, digital systems are created to fit the expectations and specifications of the design. The resulting transfer functions are subsequently applied to modify the input signals. Simulink provides an integrated environment where models can be created using a graphic user interface (GUI). Then instead of applying the input signals mathematically, you can create source objects with specifications to mimic real-life signals applicable to the system. The output signals are examined through integrated tools.

For a basic Simulink model, we need at least the following components:
Source(s): blocks that act as signal sources
Sink(s): blocks that typically act as the output.
Operator(s): blocks that applies various operations and transforms.

3. Procedures

3.1 Simple example of a Simulink model.

Launch Simulink in MATLAB by typing in "simulink" in the command window. The launch page of Simulink will appear. Create a blank model, and in this model add the following blocks by clicking "library browser", and here you can find all the needed blocks for a simple design.
From the "DSP System Toolbox", place the following blocks and specify their parameters by double click on the symbols.
Sources → sine wave: frequency 10Hz, sample time 1/1000, samples per frame 1024.
Sinks → time scope: go to file → number of input ports 3

From the "Simulink", place the following blocks:
Commonly Used Blocks → Delay: delay length → 10
Commonly Used Blocks → Gain: 0.2
Commonly Used Blocks → Sum: list of signs|++

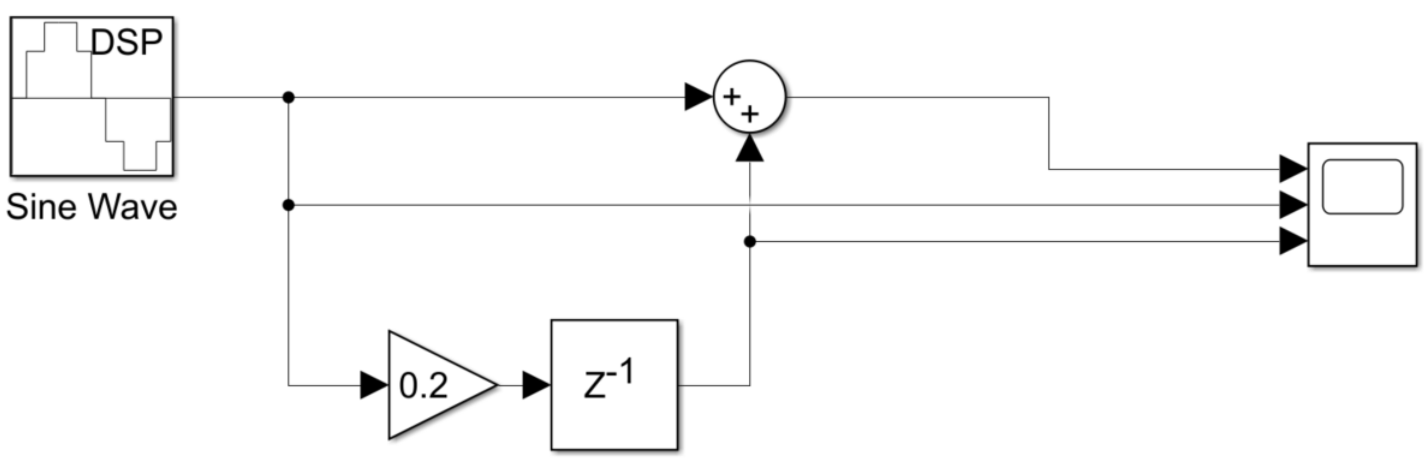
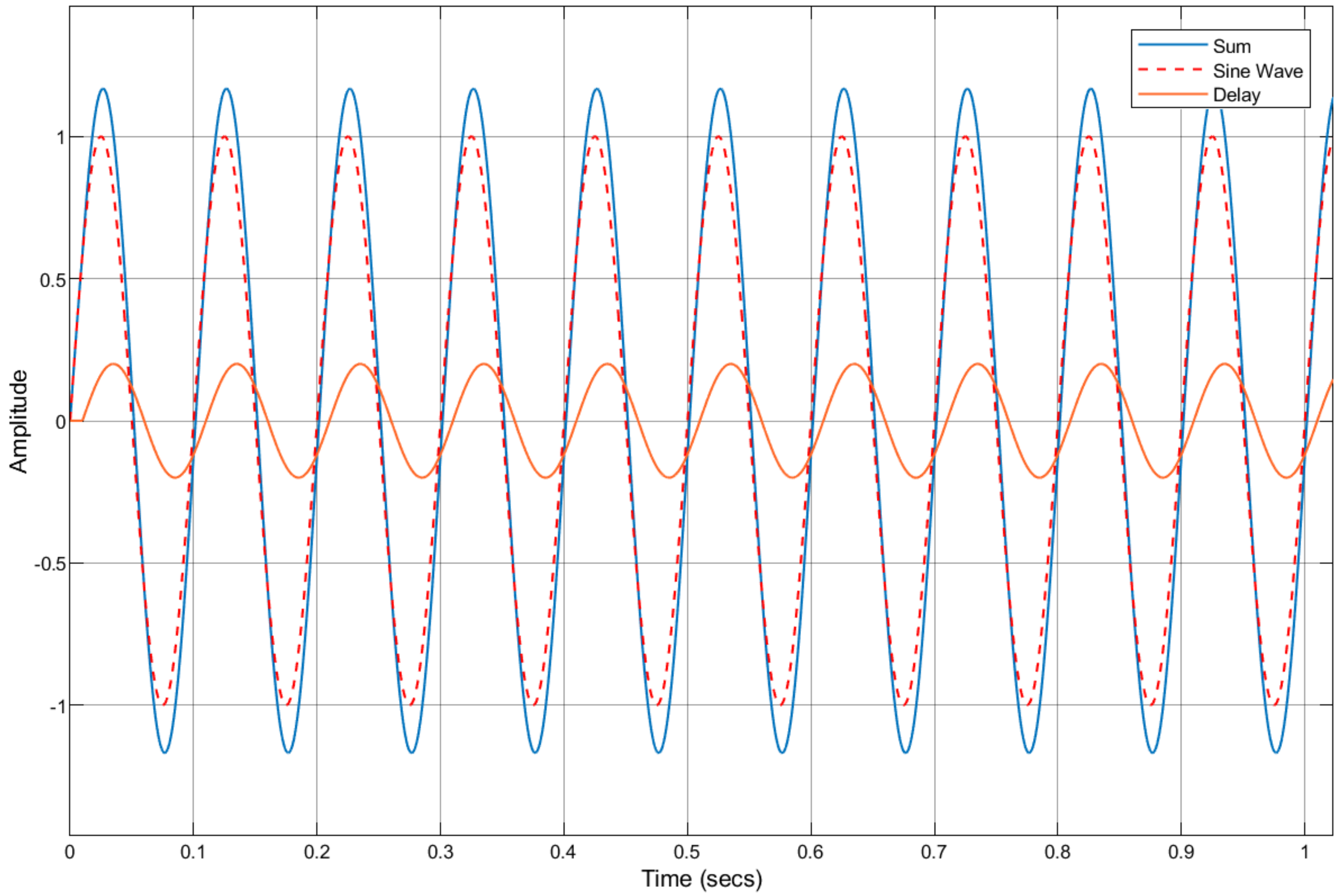


Figure 2. A simple digital system in Simulink

Task 1 (10%). Implement the following system shown in Figure 2. Once the system is completed, go to modeling at the top of the main menu and change Stop Time to 1. Press the green Run button next to it to run the simulation. Record the screenshot of the time scope with a proper range.



Task 2 (10%): Write the transfer function of this system.

$H(z) = 1 + 0.2z^{-10}$

Task 3 (10%): How to change it to make the delay to be 0.15 second?

$$\frac{0.15}{0.001} = 150$$

3.2 Implementing Simple Moving Average (SMA filter).

In our previous lab, we have implemented a SMA filter of tap 5. Now with the help of Simulink, please implement the same SMA filter. You need to change the input source signal to be Random Source. Connect the input signal and the output signal to the same time scope to view the impact of this filter.

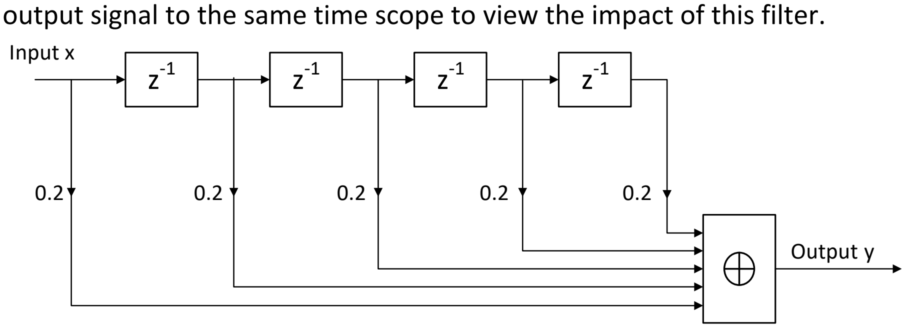
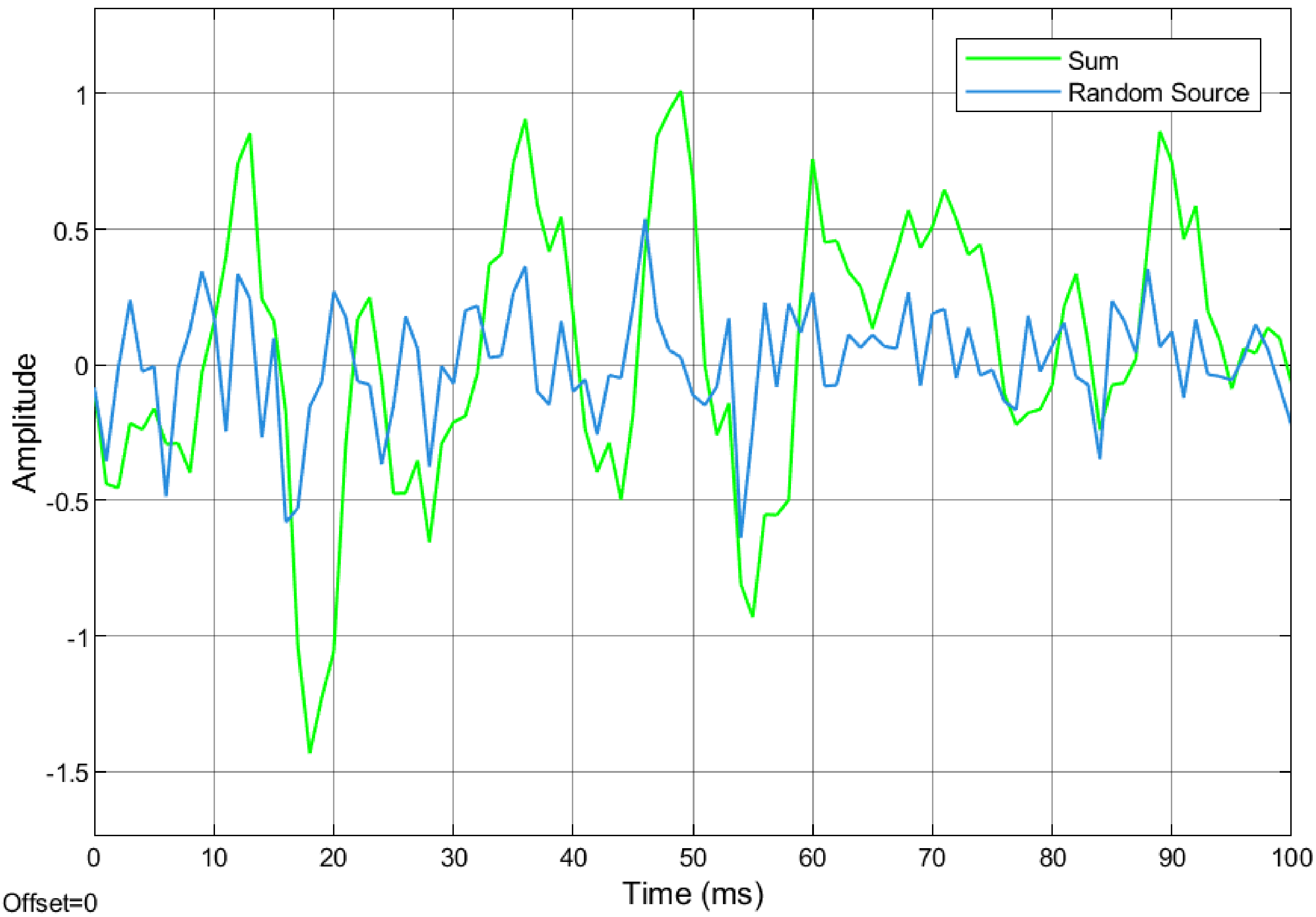


Figure 3. Block diagram of a SMA filter.

Task 4 (20%). Record the screenshots of your SMA system and the screenshot of the time scope with a proper range.



3.3 The digital filter designer tool provided by MATLAB is very useful in filter design and prototyping. Simulink provides seamless integration of this tool with its interface. You can add a filterDesigner block into your Simulink model through the following steps:
Launch Simulink Library Browser → DSP System Toolbox → Filtering → Filter Implementations and choose "Digital Filter Design".

Start a new blank model, use this procedure and add two Digital Filter Design blocks. One is high pass, the other is low pass. The specifications for each one are as shown in Figure 4.

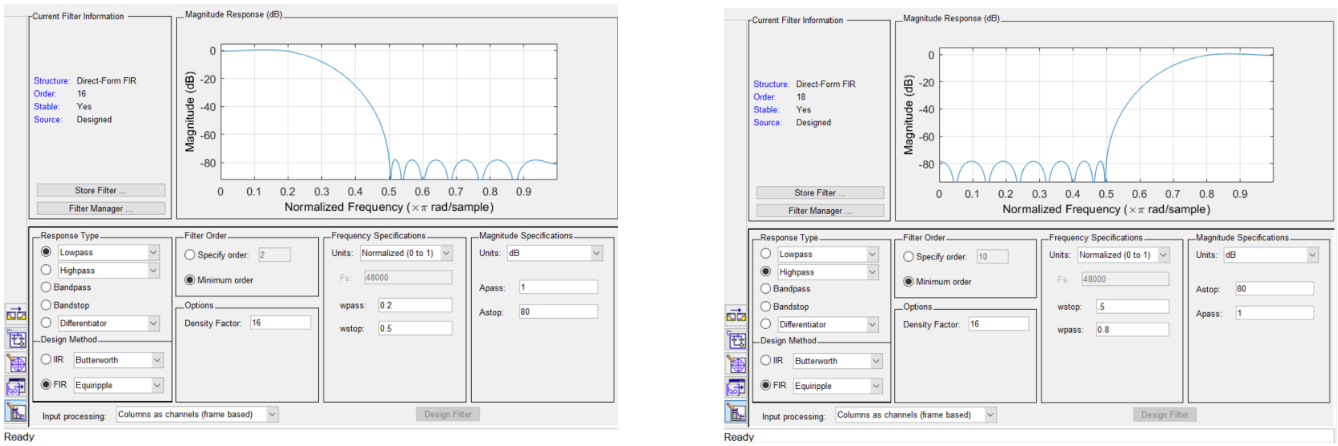
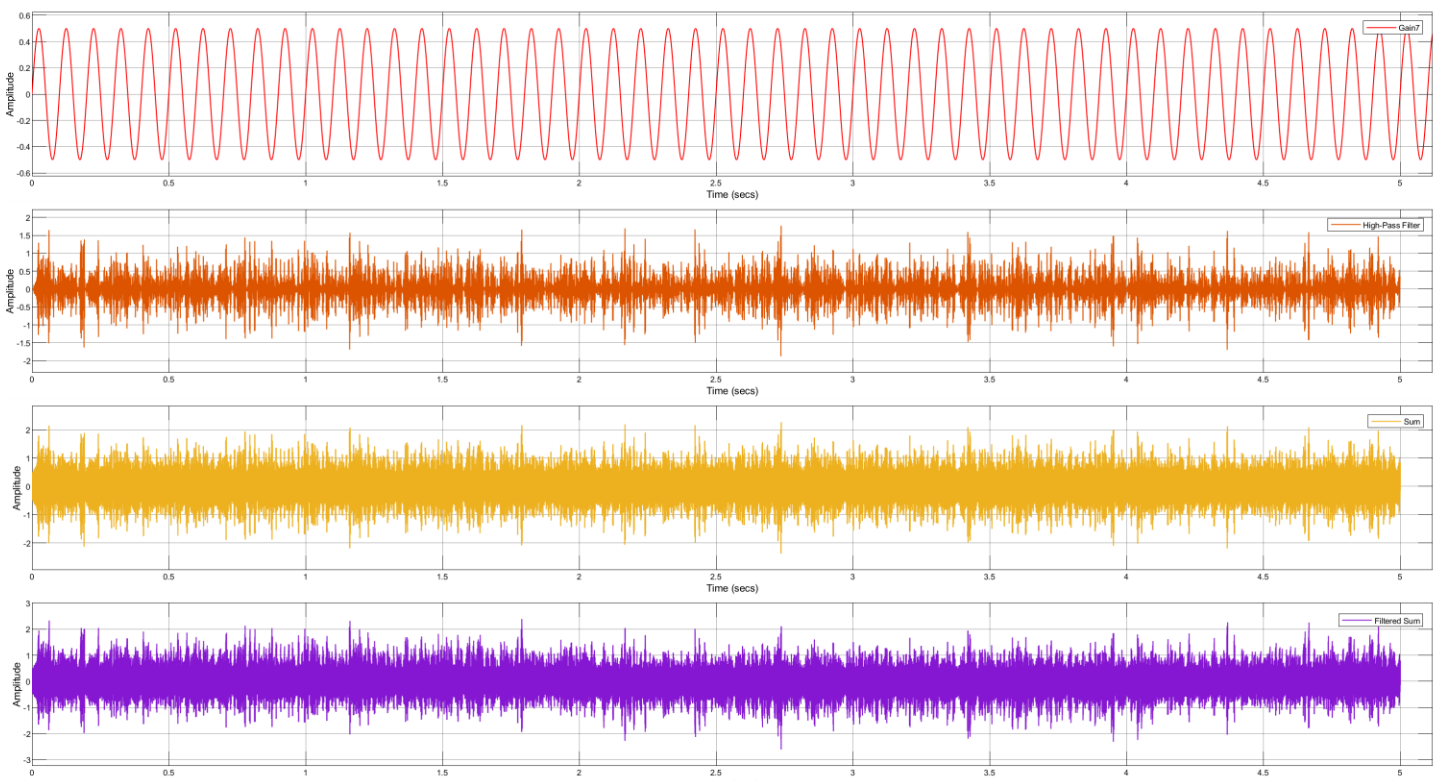


Figure 4. Specifications of low pass and high pass filter design modules.

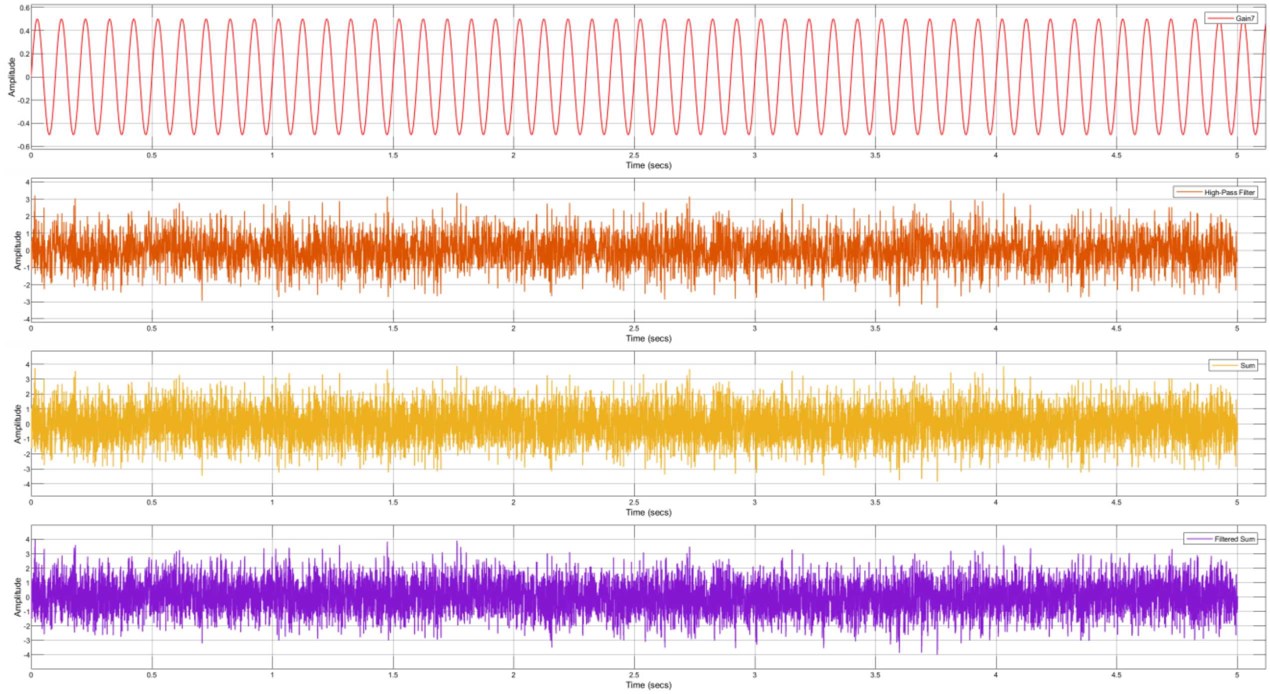
Task 5 (10%) Create a system using the High Pass and Low Pass filter modules and proper sources and sink based on the following system description:

- A random source that is filtered by the High Pass filter.
- A source of sine wave of half of the amplitude of the random source.
- Sum both source signals together and filter it using the Low Pass filter.
- Display (1) sine wave, (2) the high-pass filtered noise signal, (3) the sum, (4) filtered sum on the time scope. You need four channels for this task.

Record the screenshots of your system and the screenshot of the time scope with a proper range.

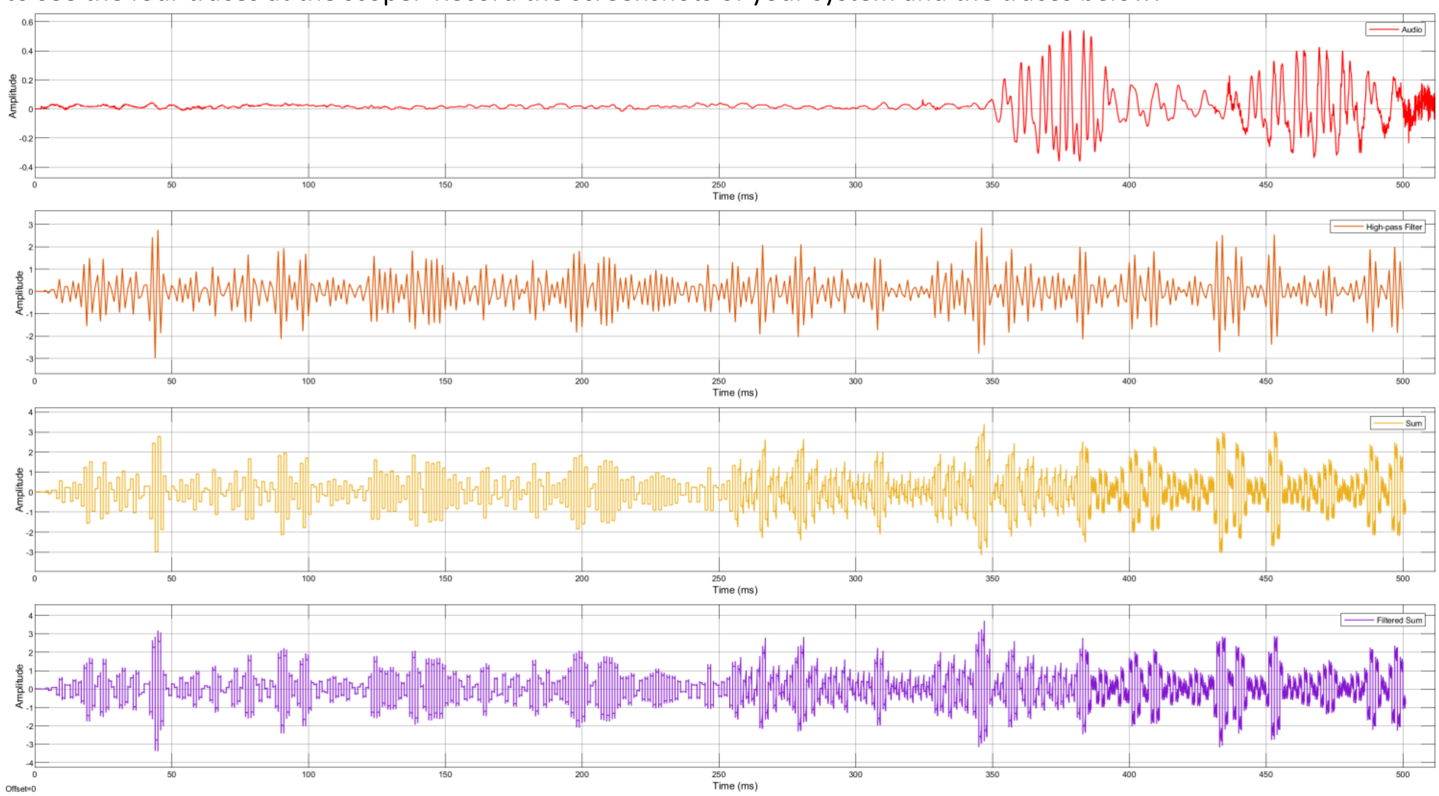


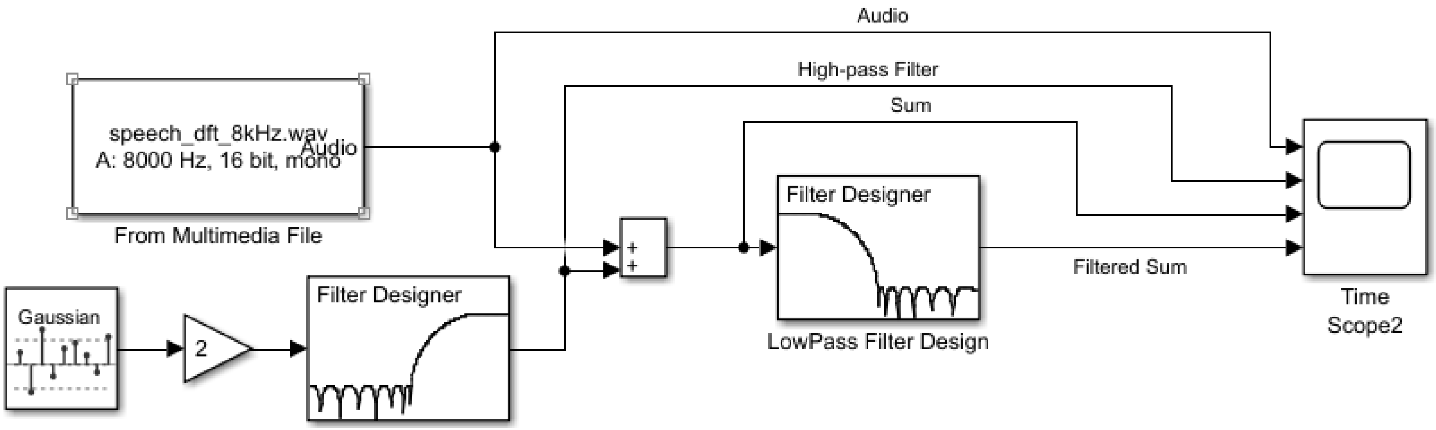
Task 6 (15%). What will happen if the High Pass filter is removed from the system? Please try it yourself and provide your explanation below and provide relevant screenshots from your simulation.



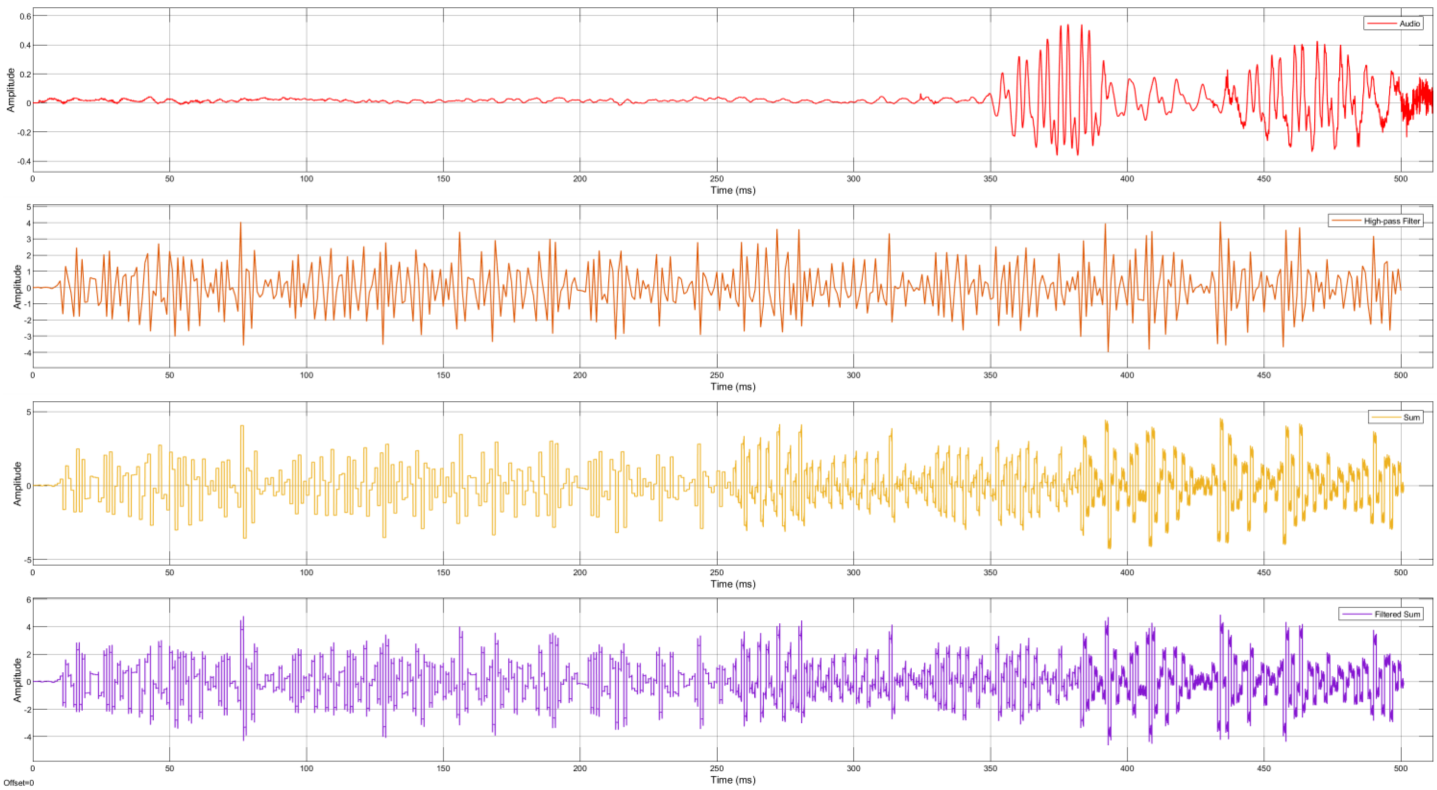
If you remove the High Pass Filter, the system will pass both high and low-frequency components, meaning more low-frequency noise will remain in the signal. Creating distortion in the frequency, causing more noise and stress on the filter. Causing the sluggish filtered sum you see instead of the smooth frequency as previously shown in Task 5.

Task 7 (10%). Now make the following change to the system to filter a real-life signal with some added noise: find the multimedia tool the "From Multimedia File" block, select a DSP sample signal such as speech_dft_8kHz.wav as the source file. Adjust the random source amplitude until is about twice as high as the speech signal. Run the whole system to see the four traces at the scope. Record the screenshots of your system and the traces below:





Task 8 (15%). What will happen if you switch the frequency specifications of the two filters? The frequency of the high pass filter is now 0.2 – 0.5, and the low pass is 0.5 – 0.8. Describe the impact of this change and provide your explanation here. Adjust the parameters of the filters so that you can see the best result, record your selected parameters in the space below.



It seems that the entire design has increased in noise and as of movement transferred over in time axis by a few milliseconds.