LAB #2 MATLAB / SIMULINK & SIGNALS

Seth Ricks

FEBRUARY 16, 2026

ECEN380-01

Table of Contents

Abstract

There are three sections to this report, separated into parts A, B, and C. Part A is simply an image of a Simulink Onramp tutorial, as a basic knowledge of Simulink was required for completing this lab. Part B works with MATLAB, testing three different systems for various properties. Part C links Simulink and a function generator together and demonstrates how sample time can affect the analyzation of a signal. This lab taught students about basic Simulink properties, testing unknown MATLAB functions for desired properties, and about sampling rate in real world signals.

Part A Procedure: Simulink Onramp

Figure 1 is an image of the certificate of completion of the Simulink Onramp training course, required to complete this lab. It included learning about the blocks in the Simulink library and connecting them together to represent a system.

MathWorks | *Training Services*

Course Completion Certificate

Seth Ricks

has successfully completed **100%** of the self-paced training course

Simulink Onramp

DIRECTOR, TRAINING SERVICES

10 February 2026

Figure 1: Simulink Onramp Certificate

Part B Procedure: MATLAB Function

In this portion of the lab, a function in a .m script called "ECEN380L2H" was provided for analysis. Included in the script were three functions. Three tests were used on each function to determine if they were linear, time invariant, memoryless, and causal.

For an input signal into the systems, a basic discrete signal was produced (See **Figure 2**). This same signal is used for all tests, since it included zero, positive, and negative values. The original input signal will be changed and mutated to satisfy the inputs for the three tests, thus the original signal is called "x_original."

```
nx = 0:19;
x_original = zeros(size(nx));
x_original(nx >= 3 & nx <= 7) = 3;
x_original(nx >= 8 & nx <= 11) = -2;
```

Figure 2: Input Used for All Proceeding Tests

Linearity Tests

For a system to be linear, it must satisfy both homogeneity and superposition. Both can be tested in one figure, by combining the test inputs into the form: $x = ax_1 + bx_2$. The constants *a* and *b* are trivial, as they are simply placeholders for the test. In this case, specific values were chosen to aid in the evaluation. A total of four signals are created: *y_separate_1, y_separate_2, y_together, and y_separate_added*. Three of these are the outputs to ax_1, bx_2, and $ax_1 + bx_2$, respectively. The final signal is the combination of *y_separate_1* and *y_separate_2* (See **Figure 3**).

```
a = 3;
b = 4;
linear_test_x1 = a * x_original;
linear_test_x2 = b * x_original;
linear_test_x_combined = linear_test_x1 + linear_test_x2;

y_separate_1 = E380L2H(nx, linear_test_x1, 1); % System 1
y_separate_2 = E380L2H(nx, linear_test_x2, 1); % System 1
y_together = E380L2H(nx, linear_test_x_combined, 1); % System 1
y_separate_added = y_separate_1 + y_separate_2;
```

Figure 3: Homogeneity and Superposition Signals for System 1

For the system to be linear, the graph of *y_together* must be identical to that of *y_separate_added.* A visualization of this is done with the MATLAB command "stem", to graph discrete-time plots. The code used to graph these inputs, along with their associated outputs, can be seen in **Figure 4** below.

```
subplot(3, 3, 1);
stem(nx, linear_test_x1(1:20),'filled','Color','b')
title('Scaled Input 1');
xlabel('nx');
ylabel('y');
grid on;

subplot(3, 3, 2);
stem(nx, y_separate_1(1:20),'filled','Color','g')
title('Output to Scaled Input 1');
xlabel('nx');
ylabel('y');
grid on;

subplot(3, 3, 4);
stem(nx, linear_test_x2(1:20),'filled','Color','b')
title('Scaled Input 2');
xlabel('nx');
ylabel('y');
grid on;

subplot(3, 3, 5);
stem(nx, y_separate_2(1:20),'filled','Color','g')
title('Output to Scaled Input 2');
xlabel('nx');
ylabel('y');
grid on;

subplot(3, 3, 7);
stem(nx, linear_test_x_combined(1:20),'filled','Color','b')
title('Scaled and Added Input');
xlabel('nx');
ylabel('y');
grid on;

subplot(3, 3, 8);
stem(nx, y_together(1:20),'filled','Color','g')
title('Output to Scaled and Added Input');
xlabel('nx');
ylabel('y');
grid on;

subplot(3, 3, 9);
stem(nx, y_separate_added(1:20),'filled','Color','r')
title('Added Outputs to Scaled Inputs 1 and 2');
xlabel('nx');
ylabel('y');
grid on;
sgtitle('\underline{System 1 Linearity Test}', 'Interpreter', 'latex');
```

Figure 4: Code for Linearity Test on System 1

The results of these subplots are depicted in **Figure 5**. As seen in this figure, *y_together* (bottom green graph) is identical to *y_separate_added* (bottom right red graph). Because of this, it can be concluded that the system **might** be linear. It is impossible to say whether this system would be linear in another test, and there are infinite tests that can be done. Because of this, the system can be assumed linear until proven that it is not.

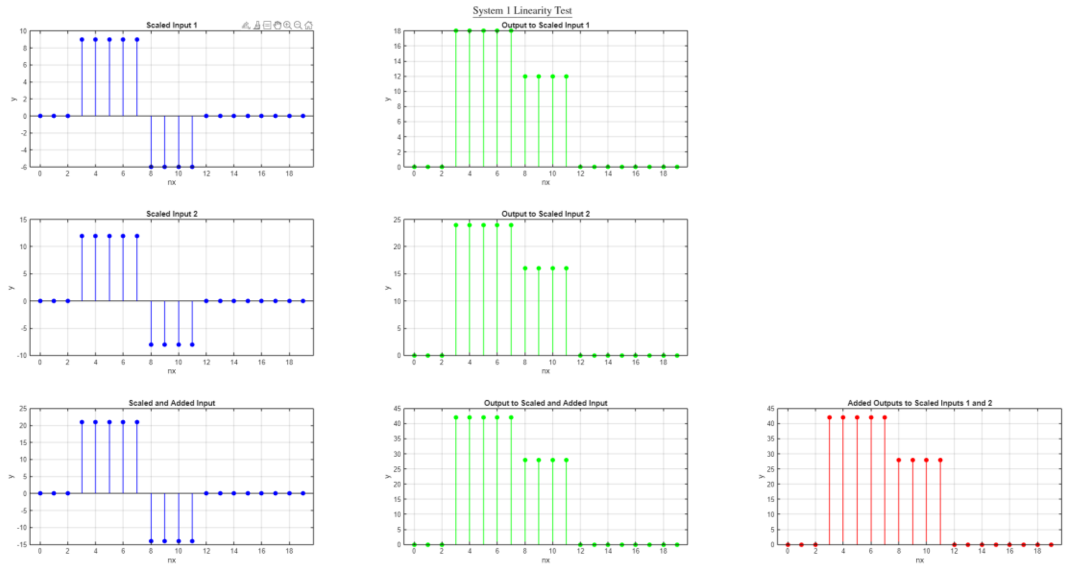

Figure 5: Results of Linearity Test on System 1

These exact same tests were also performed on systems 2 and 3, resulting in the graphs shown in **Figures 6-7** below. Due to the combined outputs of both scaled inputs being identical to the output of both those inputs combined, it can be concluded that system 2 may also be linear. This still includes the previously noted stipulation above, that it is linear until proven otherwise. For the converse reason, system 3 is deemed to be not linear.

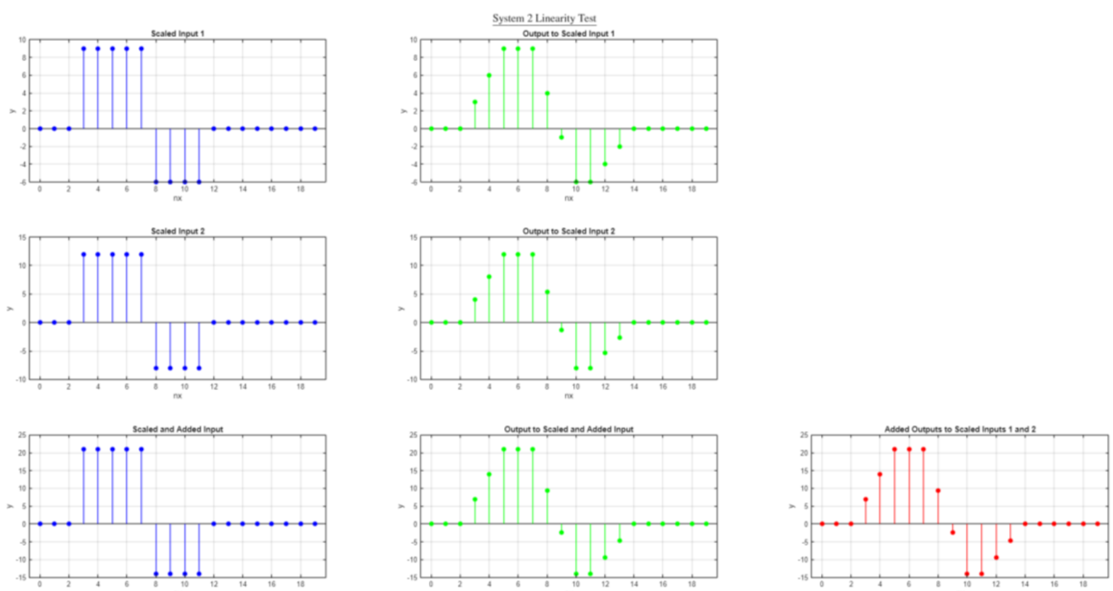

Figure 6: Results of Linearity Test on System 2

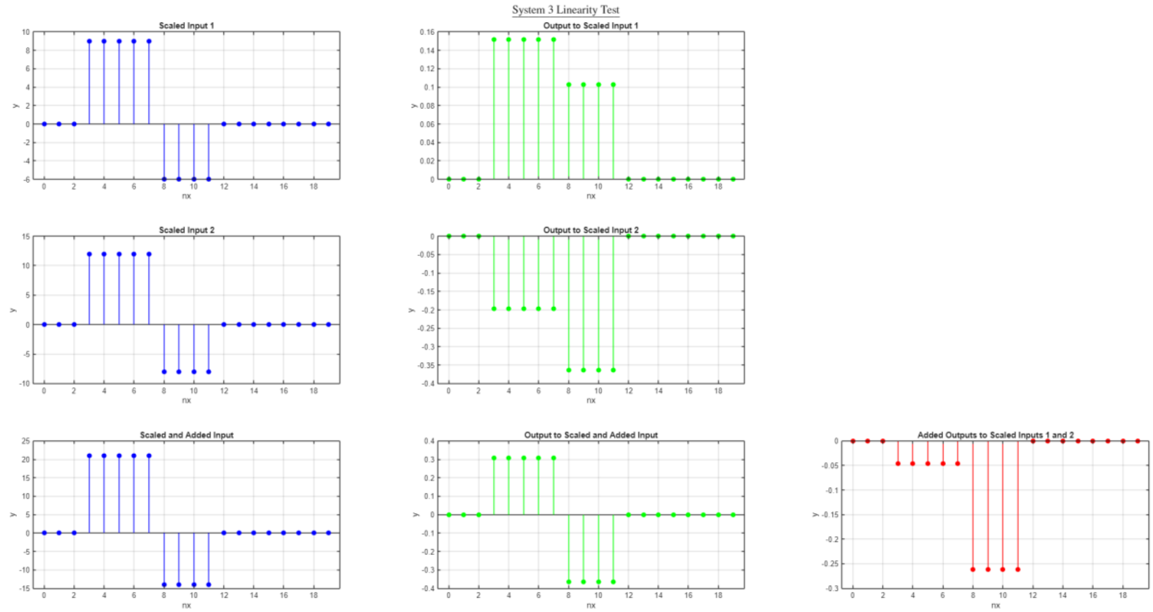

Figure 7: Results of Linearity Test on System 3

Time Variance Tests

To test if these systems are time-variant, a new input signal must be produced that is a result of shifting the original signal. The signal used in these examples can be seen in **Figure 8** below. Notice that “x_shifted” is now almost identical to the original input signal but is shifted 3 points in time to the right.

```
x_shifted = zeros(size(nx));
x_shifted(nx >= 6 & nx <= 10) = 3;
x_shifted(nx >= 11 & nx <= 14) = -2;
```

Figure 8: Shifted Input used for Time Variance Tests

For the system to be time-invariant, the original shifted output signal must be identical to the system’s response to the shifted input signal. The code for graphing this is in **Figure 9** below.

```
y_reponse_to_shift = E380L2H(nx, x_shifted, 1); % System 1
% y_reponse_to_shift = E380L2H(nx + 3, x_original, 1); % System 1

y_original = E380L2H(nx, x_original, 1); % System 1

subplot(2, 3, 1);
stem(nx, x_original(1:20),'filled','Color','b')
title('Original Input');
xlabel('nx');
ylabel('y');
grid on;

subplot(2, 3, 2);
stem(nx, y_original(1:20),'filled','Color','g')
title('Original Output');
xlabel('nx');
ylabel('y');
grid on;

subplot(2, 3, 3);
stem(nx+3, y_original(1:20),'filled','Color','r')
xlim([0 25])
title('Shifted Output');
xlabel('nx');
ylabel('y');
grid on;

subplot(2, 3, 4);
stem(nx, x_shifted(1:20),'filled','Color','b')
title('Shifted Input');
xlabel('nx');
ylabel('y');
grid on;

subplot(2, 3, 6);
stem(nx, y_reponse_to_shift(1:20),'filled','Color','r')
xlim([0 25])
title('Output Response to Shifted Input');
xlabel('nx');
ylabel('y');
grid on;
sgtitle('\underline{System 1 Time Variance Test}', 'Interpreter', 'latex');
```

Figure 9: Code for Time Variance Test on System 1

The results of the time variance tests can be seen in **Figure 10** below. As shown in the graphs, the original output shifted (top red) is identical to the output when the input is shifted (bottom red). Thus, it can be concluded that system 1 is most likely time-invariant. It is assumed to be so until proven otherwise.

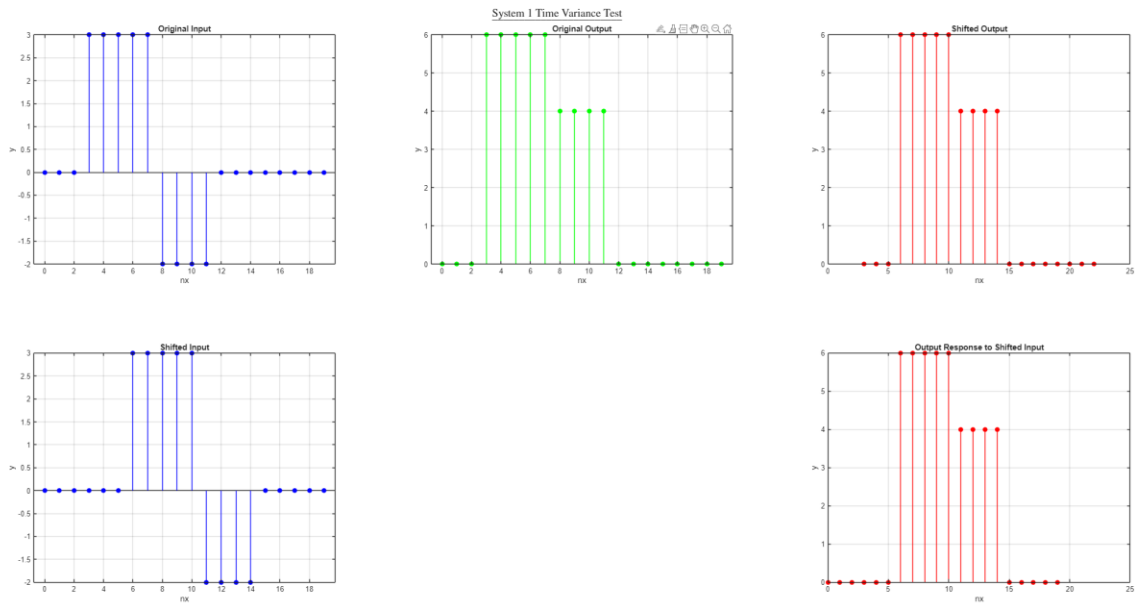

Figure 10: Results of Time Variance Test on System 1

These exact same tests were performed on systems 2 and 3, the results of which are included in **Figures 11-12**. The same conclusion as system 1 can be seen by noticing that the shifted output of the system to the original input is identical to that of the output of the system due to a shifted input of the original input. These systems are also assumed to be time-invariant until proven otherwise.

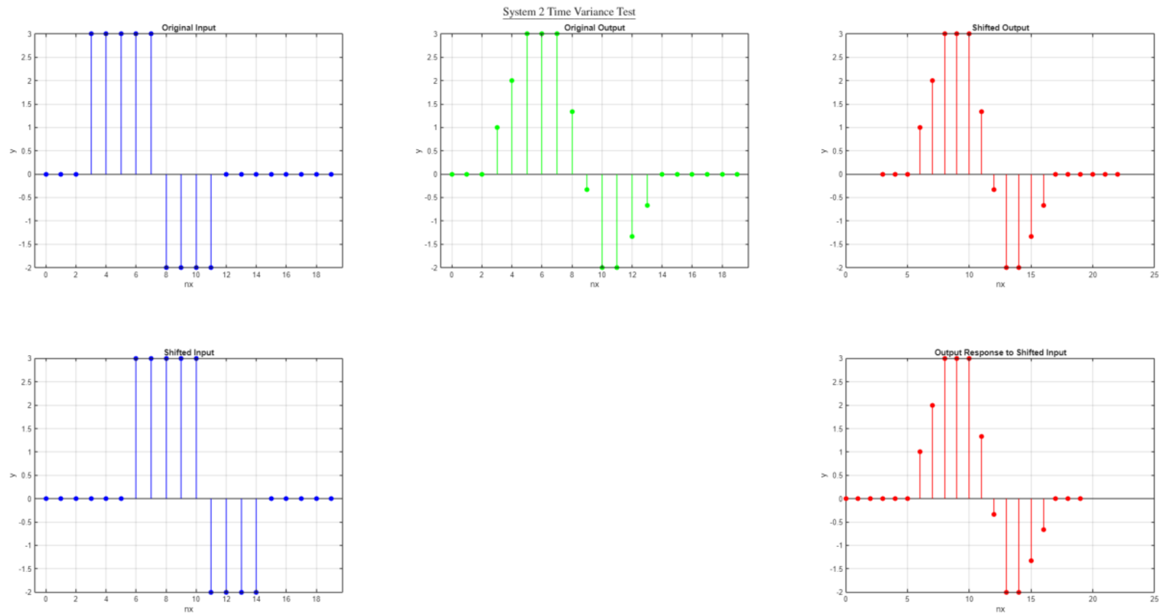

Figure 11: Results of Time Variance Test on System 2

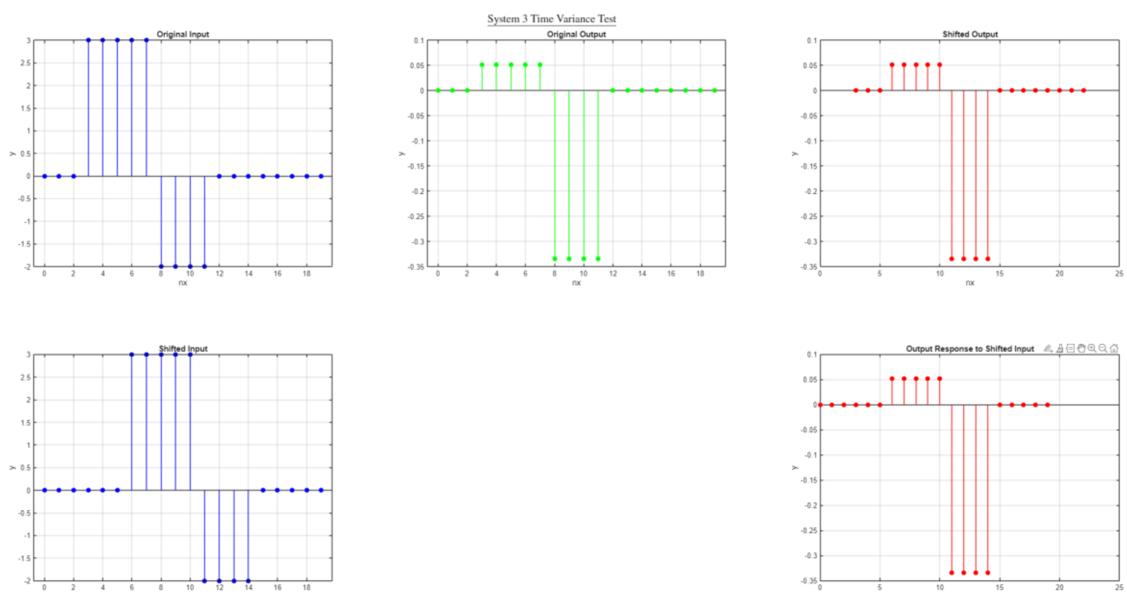

Figure 12: Results of Time Variance Test on System 3

Memoryless/Causal Tests

A system is defined as being memoryless if the output has no memory. That is, if the output includes data that is before or after the input, then it is a function of data that is either in the past or in the future. This means that it has memory. Similarly, a system is defined as being causal if the output is only a function of past or current values. Because of the similarity between these two definitions, they can be performed in the same test.

The same input as before is used, namely the “x_original” signal. This is passed into three functions, and the resulting outputs are graphed in discrete time. The code for this is simple compared to the other tests but is shown in **Figure 13** below.

```
subplot(3, 2, 3);
stem(nx, x_original,'filled','Color','b')
title('Example Input'); % Memoryless/Causal Test
xlabel('nx');
ylabel('y');
grid on;

y = E380L2H(nx, x_original, 1); % System 1

subplot(3, 2, 2);   % # rows, # columns, index
stem(nx, y(1:20),'filled','Color','r')
title('System 1 Output'); % Memoryless/Causal Test
xlabel('nx');
ylabel('y');
grid on;

y = E380L2H(nx, x_original, 2); % System 2

subplot(3, 2, 4);
stem(nx, y(1:20),'filled','Color','b')
title('System 2 Output'); % Memoryless/Causal Test
xlabel('nx');
ylabel('y');
grid on;


y = E380L2H(nx, x_original, 3); % System 3

subplot(3, 2, 6);
stem(nx, y(1:20),'filled','Color','g')
title('System 3 Output'); % Memoryless/Causal Test
xlabel('nx');
ylabel('y');
grid on;
sgtitle('\underline{Systems 1-3 Memoryless/Causal Tests}', 'Interpreter', 'latex'); % Overall Title
```

Figure 13: Code for Systems 1-3 Memoryless/Causal Tests

Because of the simplicity of these tests, all three results can be seen in one graph (See **Figure 14**). Because the output only resides at the same time as the input, they can be deemed to be memoryless and causal in the same test. They are assumed to be so until proven otherwise.

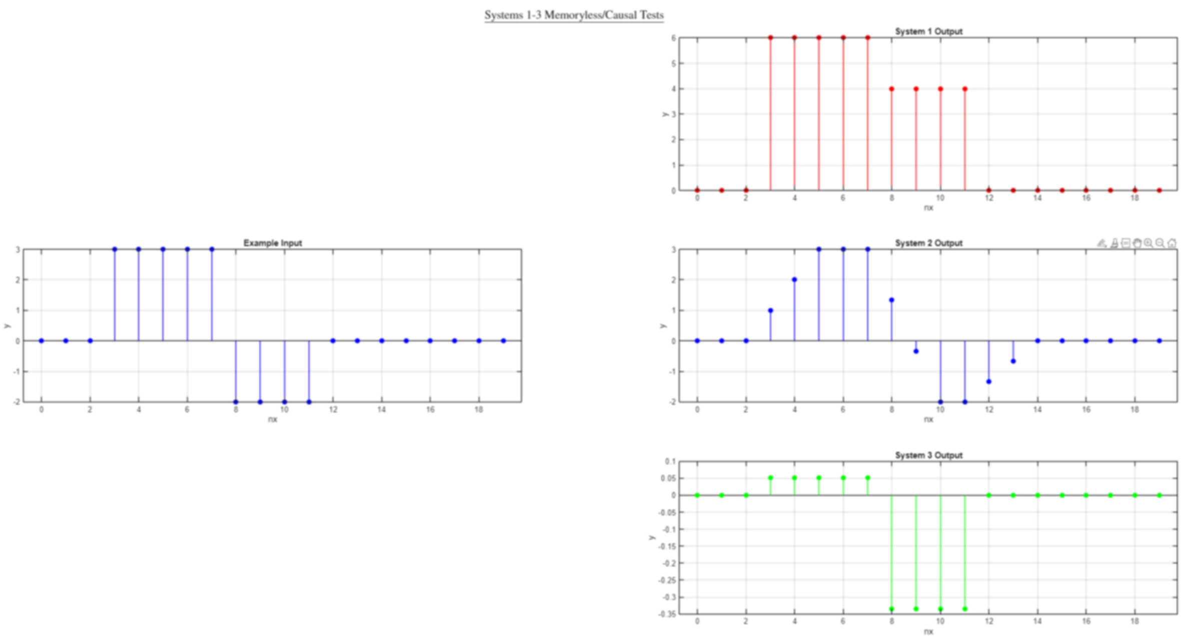

Figure 14: Results of Memoryless/Causal Tests on Systems 1-3

The results of all the tests on systems 1-3 can be seen in the table in **Table 1** below.

	System 1	System 2	System 3
Linear	yes	yes	no
Time-Variant	yes	yes	yes
Memoryless	yes	yes	yes
Causal	yes	yes	yes

Table 1: Results Summary of All Tests

Part C Procedure: MATLAB Analog Scope

For this portion of the lab, the PCIe-6321 National Instruments board was used. This was used as a system, with an input and an output. Testing different input frequencies using a function generator, the outputs and the inputs were graphed. The point of the system was to simply try to replicate the input as closely as possible, with some errors with some frequencies. This was due to a sample time of 10ms, explained later. Images of oscilloscope measurements, as well as Simulink scope images are included.

First the system was represented as a simple block diagram in Simulink (See **Figure 15**). The PCIe-6321 board was selected for both the analog input and the analog output. A scope is connected to the node in between them, to visually graph the result.

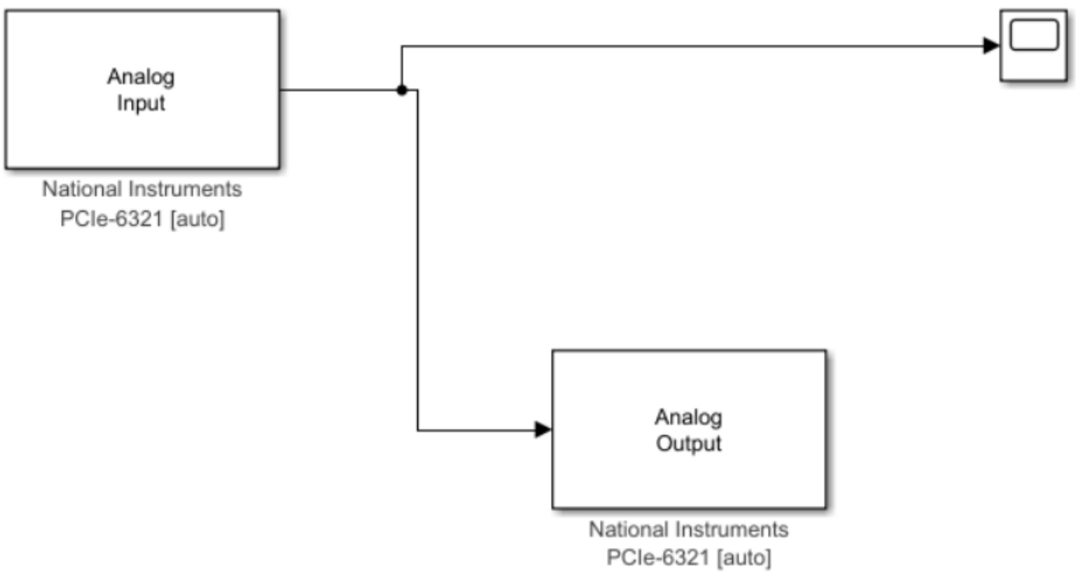

Figure 15: Simulink Block Diagram Representation

A single-ended connection was wired on the physical board, which was then connected to the computer tower. Using a function generator, sinusoidal waves with peak-to-peak voltages of 1V and frequencies of 2Hz, 20Hz, 200Hz, 500Hz, and 1002Hz were generated. The results of these tests can be seen in the next several pages, namely **Figures 16-30** below. **Table 2** is given as a reference. The results will be analyzed afterwards.

	2Hz	20Hz	200Hz	500Hz	1002Hz
Oscilloscope Image	Figure 16	Figure 19	Figure 22	Figure 25	Figure 28
Simulink	Figure 17	Figure 20	Figure 23	Figure 26	Figure 29
Enlarged Simulink	Figure 18	Figure 21	Figure 24	Figure 27	Figure 30

Table 2: Frequency Results

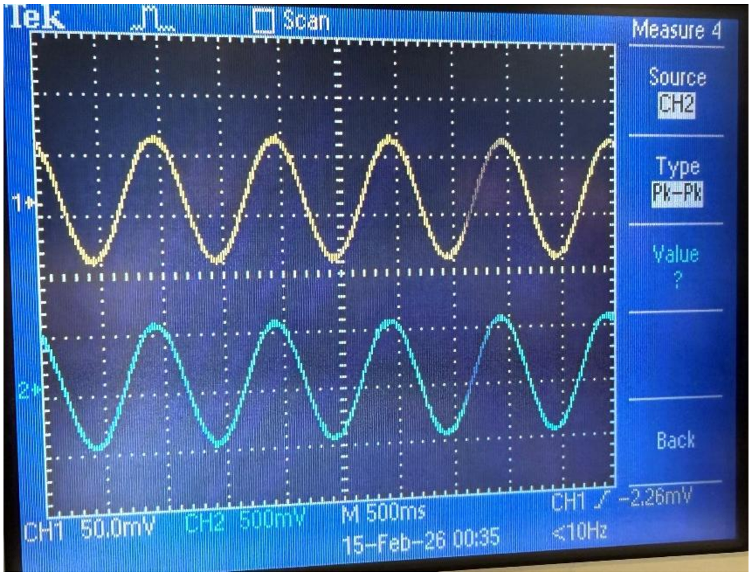

Figure 16: 2Hz Oscilloscope

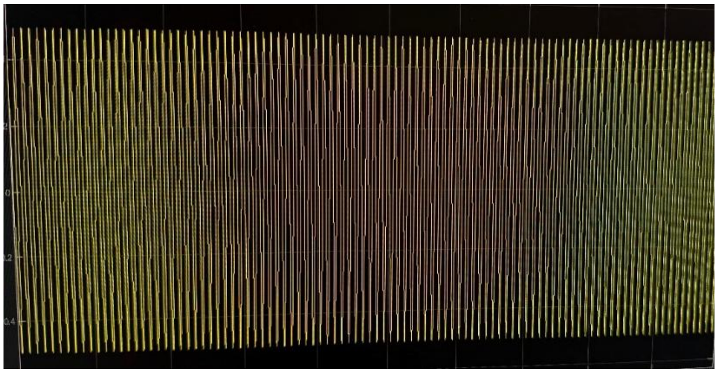

Figure 17: 2 Hz Simulink

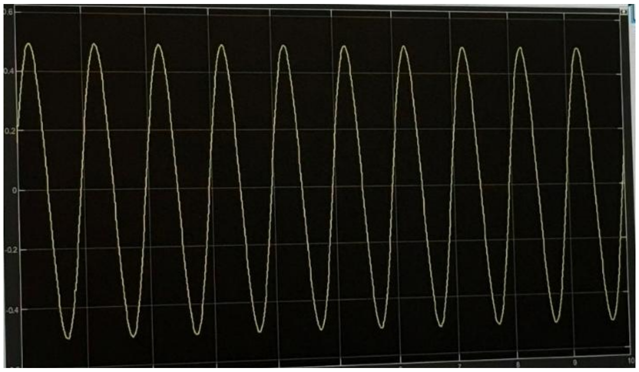

Figure 18: 2Hz Enlarged Simulink

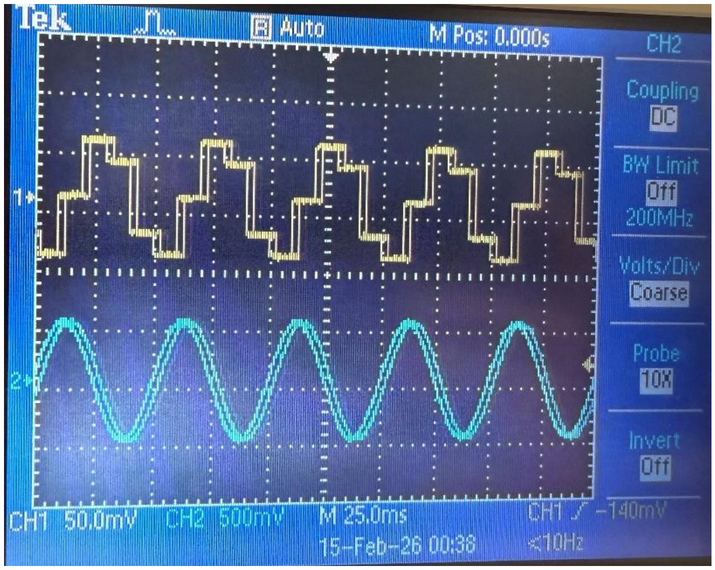

Figure 19: 20Hz Oscilloscope

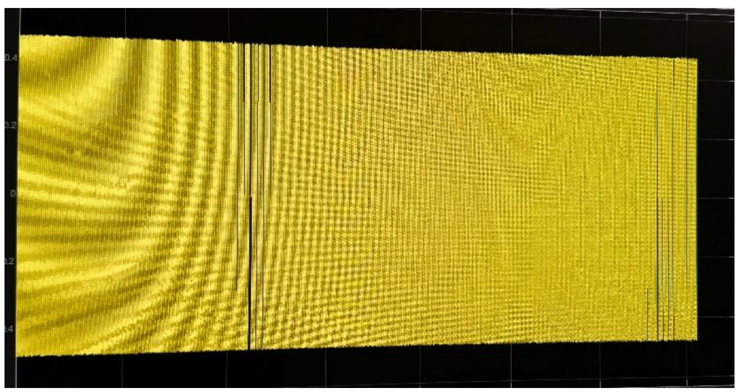

Figure 20: 20 Hz Simulink

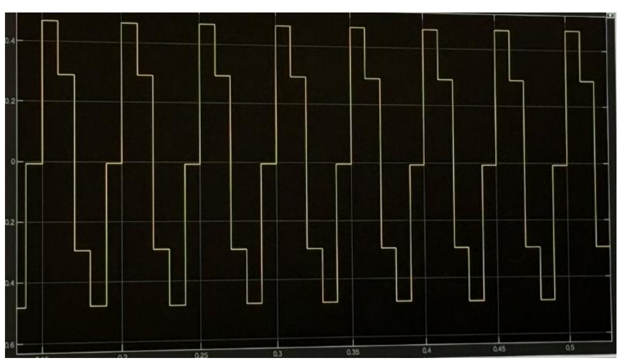

Figure 21: 20Hz Enlarged Simulink

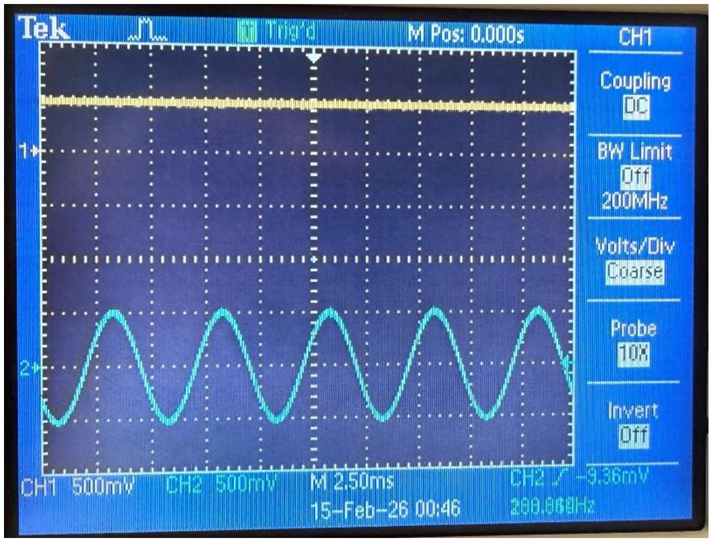

Figure 22: 200Hz Oscilloscope

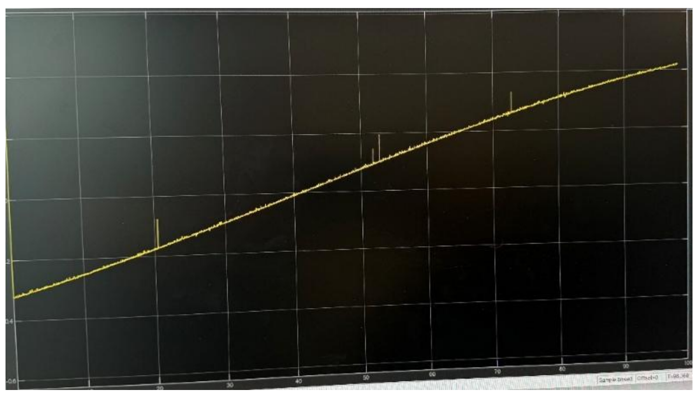

Figure 23: 200Hz Simulink

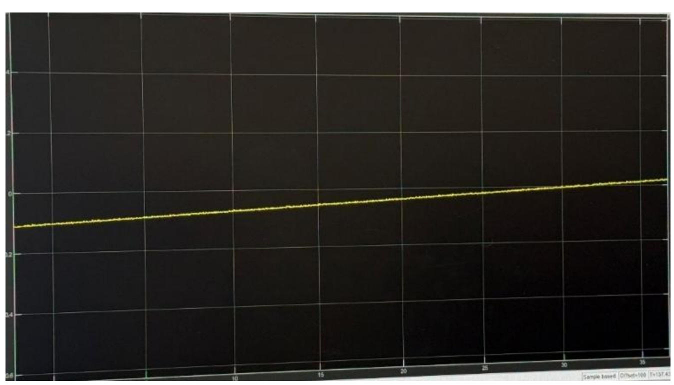

Figure 24: 200Hz Enlarged Simulink

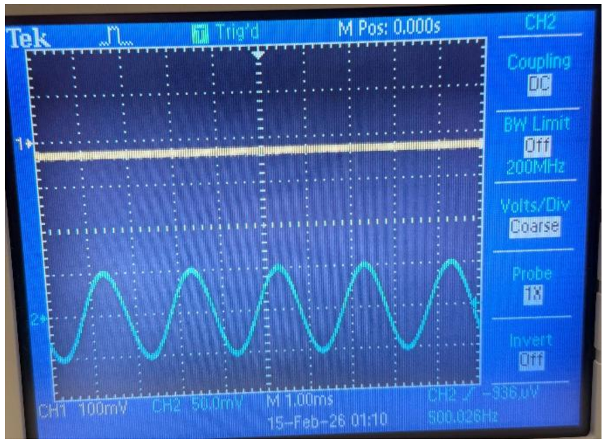

Figure 25: 500Hz Oscilloscope

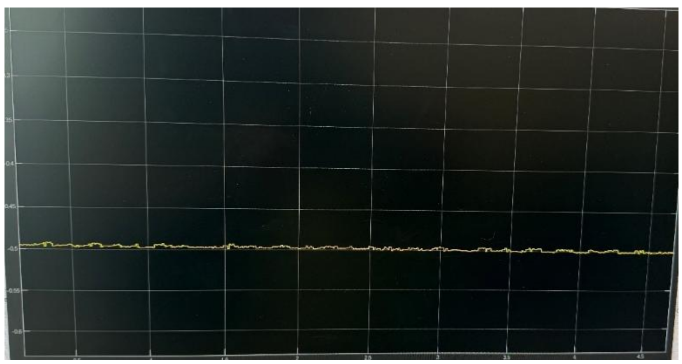

Figure 26: 500Hz Simulink

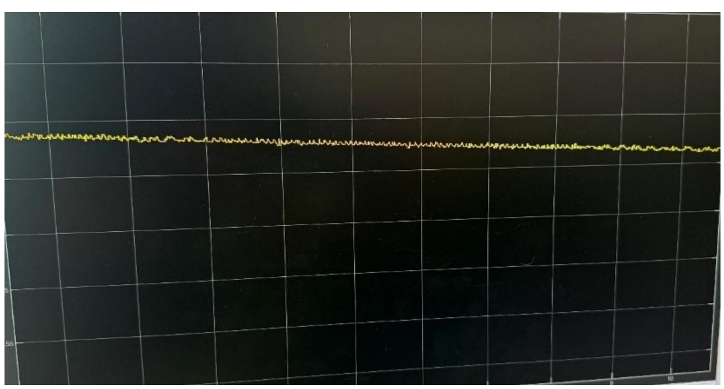

Figure 27: 500Hz Enlarged Simulink

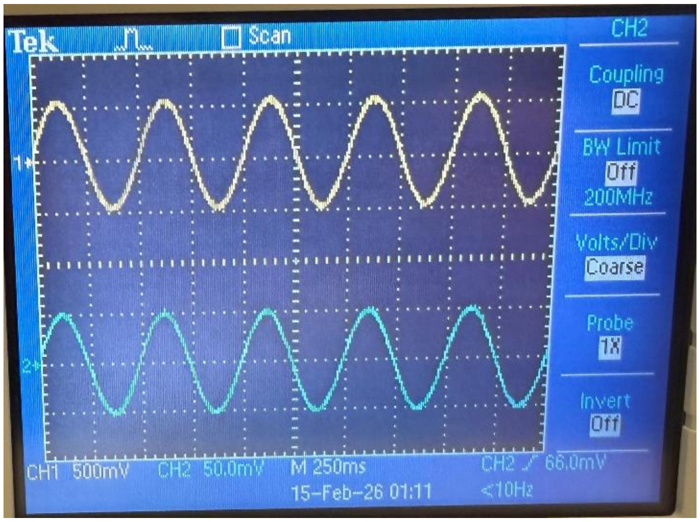

Figure 28: 1002 Hz Oscilloscope

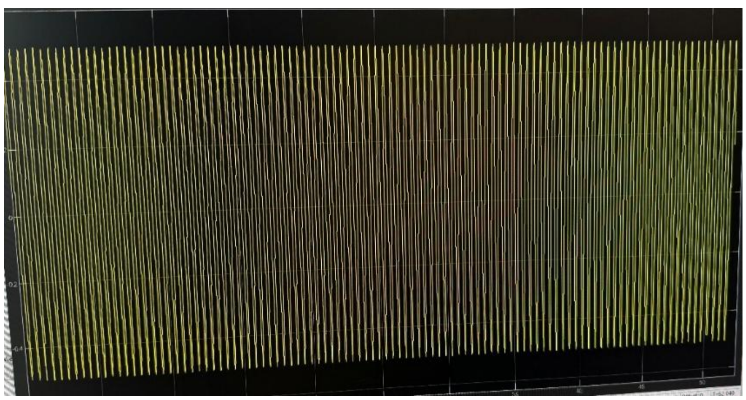

Figure 29: 1002Hz Simulink

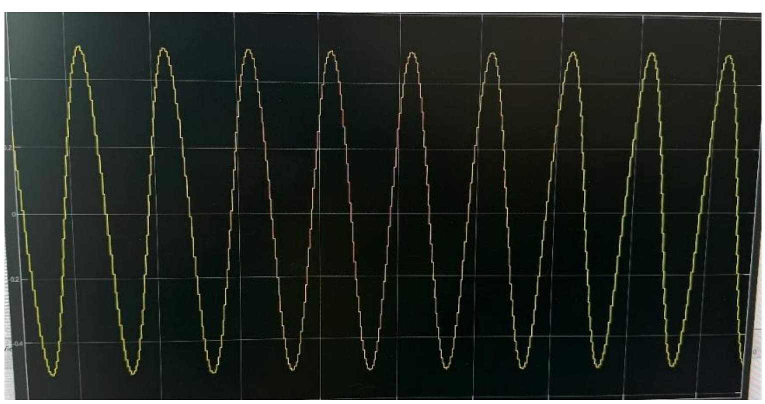

Figure 30: 1002Hz Enlarged Simulink

To aid in the analyzation of these results, a few questions were asked:

Question: Can you still measure the correct period at each frequency in Simulink? How or why not?

Answer: No, not all the frequencies can be measured correctly in Simulink. For example, the 200Hz and 500Hz frequencies do not appear to be sinusoidal at all when zoomed in. Their measured frequencies are so low that they appear to be flat lines when enlarged.

Question: Try to explain why the 1002Hz signal looks the way it does in Simulink.

Answer: This is due to the Stroboscopic Effect. This effect is when sampling is done at the same rate or a multiple of the rate of the input frequency, and results in something unreliable. This is because it is sampling at the same point in the signal. Due to a sampling period of 0.01 seconds, or a frequency of 100Hz, anything that is equal to or a multiple of 100Hz fails to be represented correctly in Simulink. This is why 200 and 500Hz failed, but 2, 20, and 1002Hz worked.

Question: Could we recover the original waveform at each frequency? How or why not?

Answer: No, for the reasons given above. If the frequency of the input is not equal to or a multiple of 100Hz, it is easy to discern the original frequency. It is impossible otherwise.

To test the theory that the reason for the signal representation failure was due to the stroboscopic effect, the input was slowly changed from 999Hz to 1002Hz, instead of directly setting it to the latter. 999 Hz was measured accurately, but as soon as the frequency changed to 1kHz, the Simulink scope flatlined. When the input frequency changed to 1001 and 1002 Hz, the scope returned to accurately depicting the input frequency. This confirmed the theory of the error.

This effect is also the reason that ceiling fans appear to change direction, and that car tires appear to be moving slower than they are. The human eye can't sample at the speed of rotation that the object is moving, so it captures something that isn't occurring. To visualize the stroboscopic effect in more detail, see this website: stroboscopic effect link

Conclusion

In this lab, a basic knowledge of Simulink and how to use it was gained and applied. MATLAB skills were also built upon to better visualize memoryless, causal, time-invariant, and linear systems. The greatest thing that was learned was how a fixed sampling rate can affect the result of an input signal with the same rate, known as the stroboscopic effect.

The most significant struggle in this lab was finding a computer with both the correct hardware and software to complete Part C. Many of the computers in the Science and Technology Center at BYU-Idaho didn't have either. The software couldn't simply be installed in a machine with the correct hardware because of admin restrictions. Finding one that was compatible with both the hardware and the software necessary took a couple of hours. But once a viable machine was found, the rest of the lab was not too challenging.

In future labs and experiences, it will be worthwhile noting the frequency of which a system is sampling its input. This may affect the output if the input frequency is the same or a multiple of the sampling rate.

As a note for improvement of this lab, it would be much easier for the instruction document to be in a linear fashion, instead of jumping around from section to section. The way that it currently is, it was very easy to be lost in what step is currently being completed.