Assignment 2: LTI System Responses and Convolution

Team 19 - Andrew De Rango, Garret Langlois, Ethan Otteson
IBEHS 3A03
Biomedical Signals and Systems
Fall Term 2024

Table of Contents

Introduction

In this assignment, students explore the properties of linear time-invariant (LTI) systems. This included determining the unit impulse and unit step responses and their relationship through the cumulative summation and first difference for provided LTI systems. Additionally, two real biomedical signals are provided and the LTI systems are explored for their relation to the convolution operation. This report touches on each of these concepts and provides a graphical analysis and formal logical test when useful. Bonus 1 implements formal logical tests and Bonus 2 explores the filtering properties of each LTI system.

The following table outlines all the files submitted and their purpose.

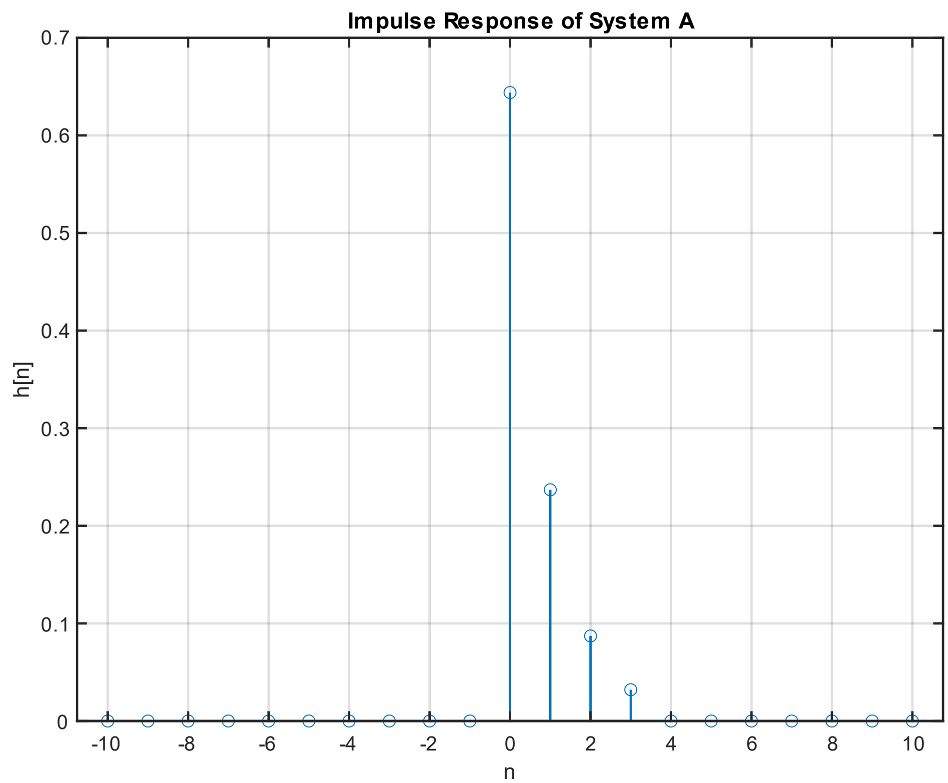
Filename	Description
main.m	A single file responsible for running tests on systems A, B, and C. It is broken up into the following functions: I. **calcImpulseResponse()** Returns the impulse response of the system and has optional plotting. II. **calcStepResponse()** Returns the unit step response of the system and has optional plotting. III. **calcCumSumImpulseResponse()** Returns the cumulative sum of the impulse response of the system and has optional plotting. IV. **calcStepResponseDiff()** Returns the first difference of the unit step response of the system and has optional plotting. V. **CalcECGResponse()** Returns the output of the system when provided the ECG signal. VI. **CalcRespirationResponse()** Returns the output of the system when provided the respiration signal VII. **verifyConvolution()** Plots both the convolution of the impulse response of the systems with the ECG and Respiration inputs and the direction computation of the system's response to the Respiration and ECG signals. Returns a Boolean value if they are the same to check in the logical test function. VIII. **formalLogicalTest()** Uses a tolerance value to check for equality between the two comparators in III, IV, and VII. IX. **filterTest()** Performs a Fourier Transform on the system's impulse responses and plots the frequency spectrums of each.

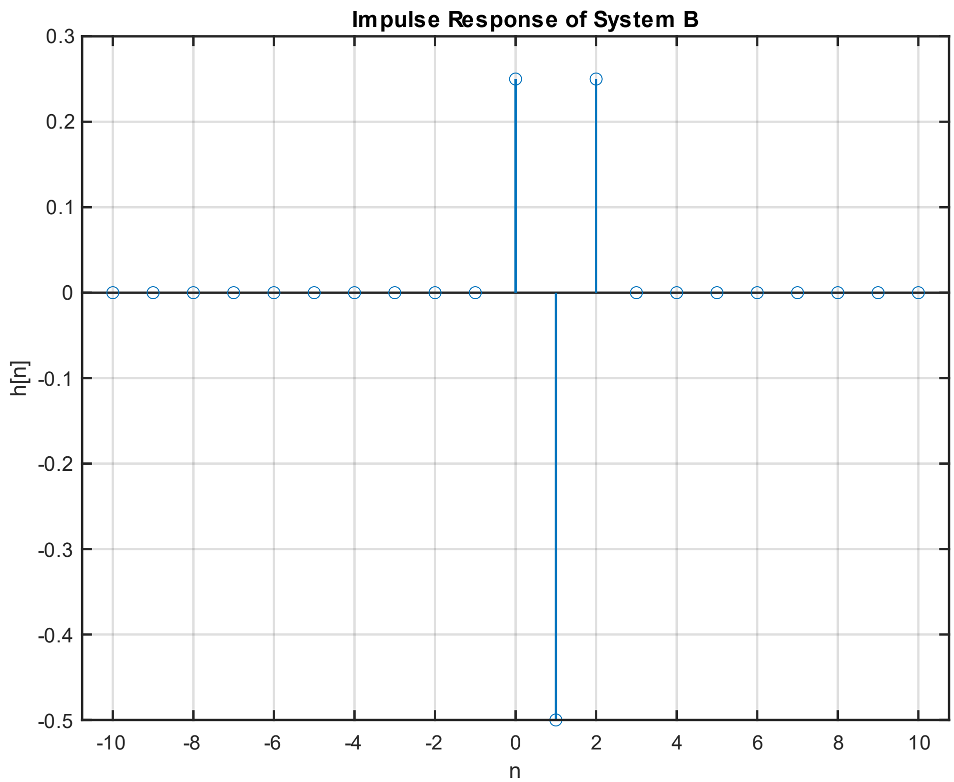
Unit Impulse Response

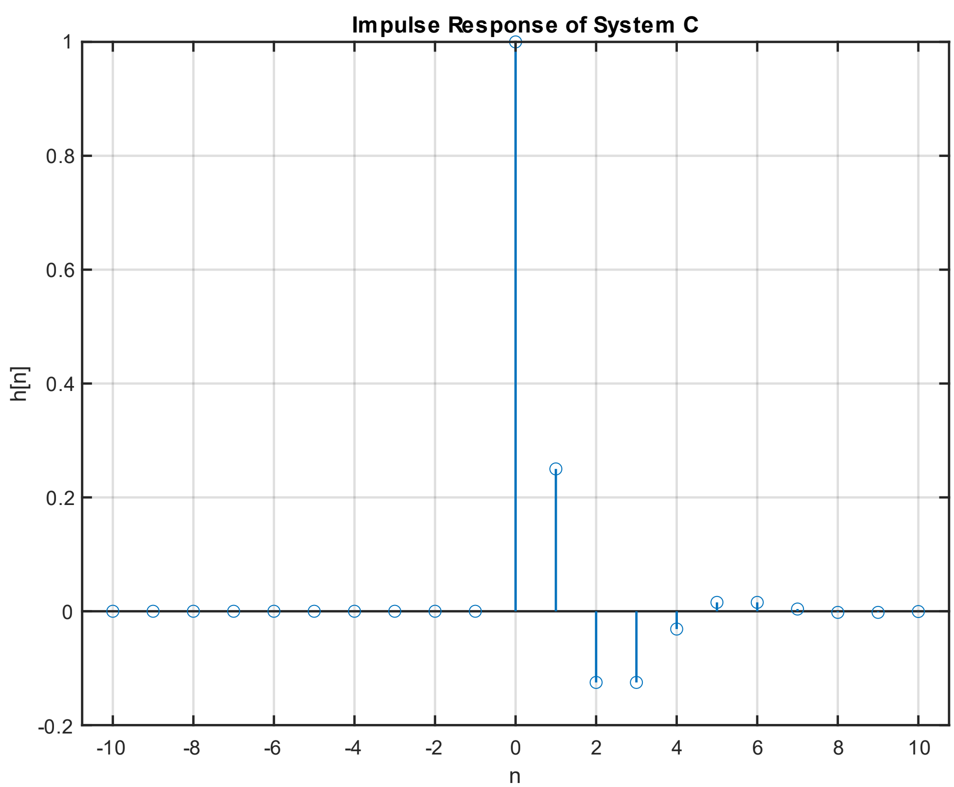
Part I of the assignment is to compute the system's output when the input is the unit impulse function $\delta[n]$. The output in this circumstance, otherwise known as the impulse response, can be valuable especially for LTI systems in characterizing the system's behaviour in both the time domain and the frequency domain. This will be explored in the following sections of the assignment.

To compute the unit impulse response, the unit impulse function $\delta[n]$ was constructed by forming a vector of length determined by the length of the n range, with the index $n = 0$ being assigned a value of 1. Then, this input signal is put through the system. If the parameter $showPlot$ is true, then the following plots may be obtained, demonstrating the impulse responses $h[n]$ for System A, System B, and System C.

Observing that the impulse response for System A and System B appear to decay to 0 in a finite number of samples in both the positive and negative directions, it should be noted that the first two systems each have a finite impulse response (FIR). To validate the visual observations, we can enable the verbose option for the first function and determine that for System A, $h[n] = 0$ for all $n \geq 4$ and $n < 0$. For System B, $h[n] = 0$ for all $n \geq 3$ and $n < 0$. However, System C has an infinite impulse response (IIR) because it never completely decays to 0 before ∞ in the positive direction despite having the appearance of converging to 0. This can be supported numerically if we modify the n range and find $h[1000]$, which renders approximately $-4.6663 \cdot 10^{-302}$; a value incredibly close to 0 but not exactly. Thus, it is reasonable to assume that the impulse response $h[n]$ for System C will continuously but asymptotically approach 0, effectively rendering an IIR.



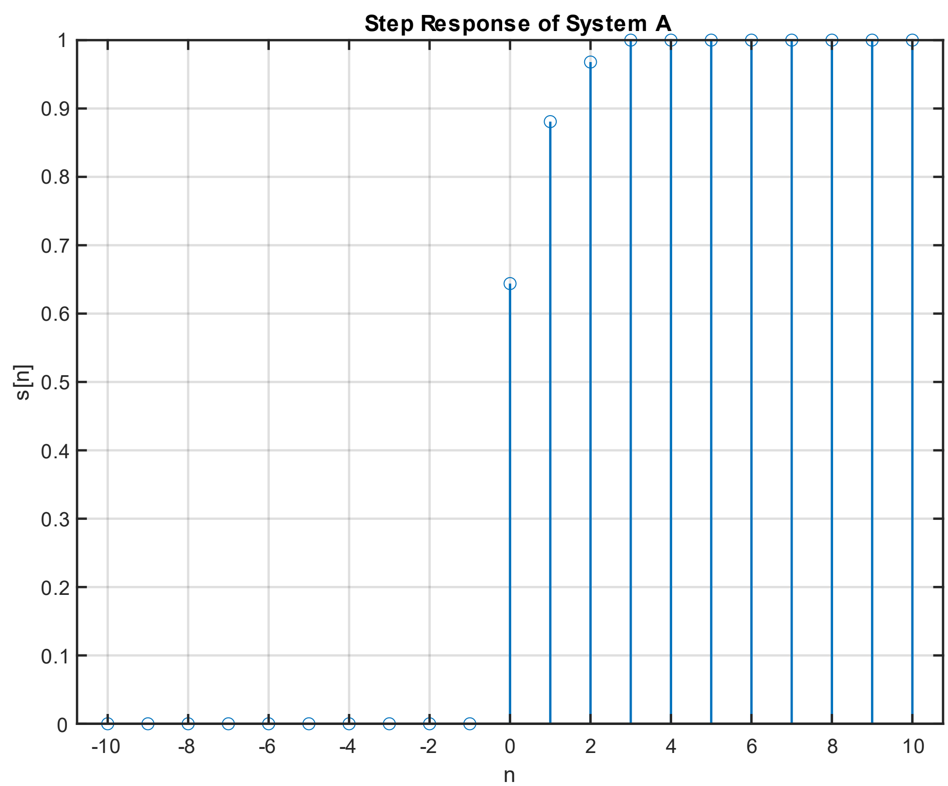

Impulse Response of System B
h[n]
n
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-10
-8
-6
-4
-2
0
2
4
6
8
10

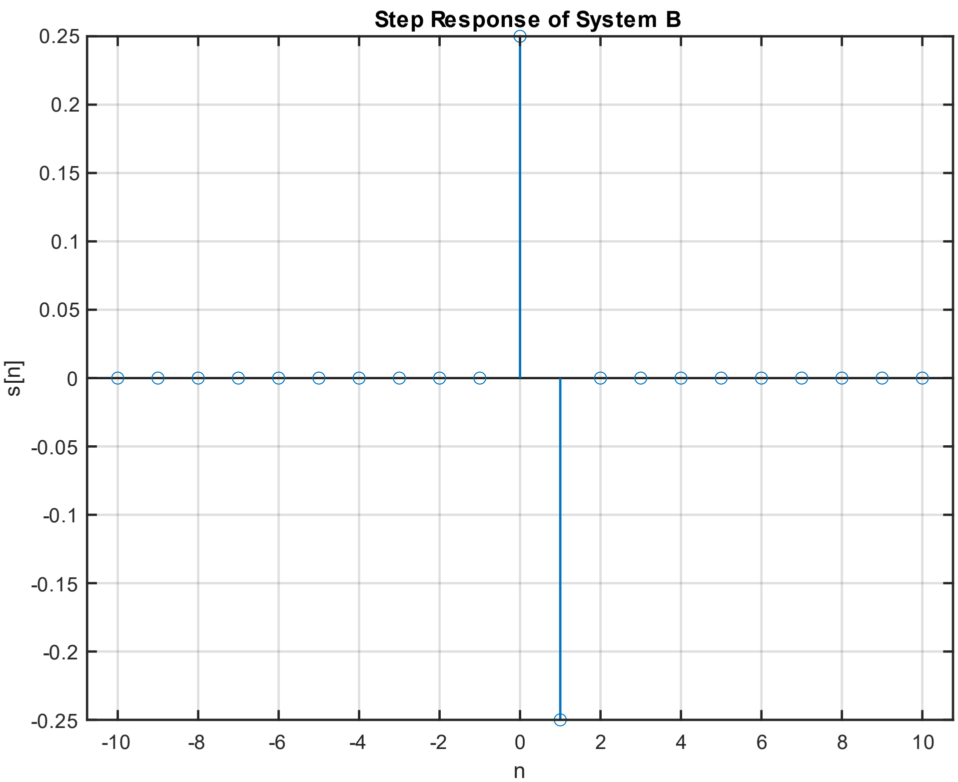

Impulse Response of System C
h[n]
n
1
0.8
0.6
0.4
0.2
0
-0.2
-10
-8
-6
-4
-2
0
2
4
6
8
10

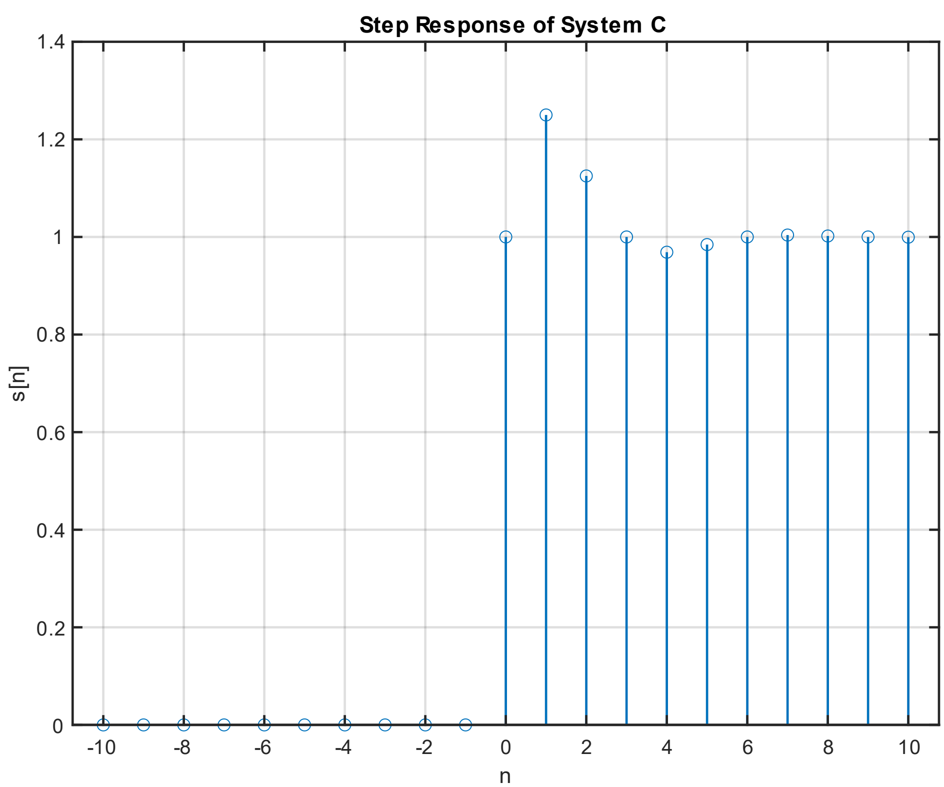
Unit Step Response

Part II of the assignment is to compute the system's output when the input is the unit step function $u[n]$. The output in this circumstance, otherwise known as the step response, can be valuable especially for LTI systems in characterizing the system's behaviour in both the time domain and the frequency domain.

To compute the unit step response, the unit step function $u[n]$ was first constructed by forming a vector of length determined by the magnitude of n, with indices $n \geq 0$ being assigned a value of 1. Then, this input signal is put through the system. If the parameter $showPlot$ is true, then the following plots may be obtained, demonstrating the step responses $s[n]$ for System A, System B, and System C.




Step Response of System B
s[n]
n


Step Response of System C
s[n]
n

Cumulative Sum

Part III of the assignment requires showing that the output from the unit step function $u[n]$, as derived from Part II, is equivalent to the cumulative sum of the impulse response $h[n]$, as can be derived from the result of Part I.

This relationship can be understood through the properties of LTI systems. It is worth considering that the cumulative sum is the discrete analog of integration with bounds $-\infty$ to t. In continuous time, the following property holds, as starting from the inputs:

$$u(t) = \int_{-\infty}^{t} \delta(\tau) \cdot d\tau$$

Converting to the analogous discrete formulation, it can be stated and verified that the unit step function is equivalent to the cumulative sum of the unit impulse function, as so:

$$u[n] = \sum_{i=-\infty}^{n} \delta[n]$$

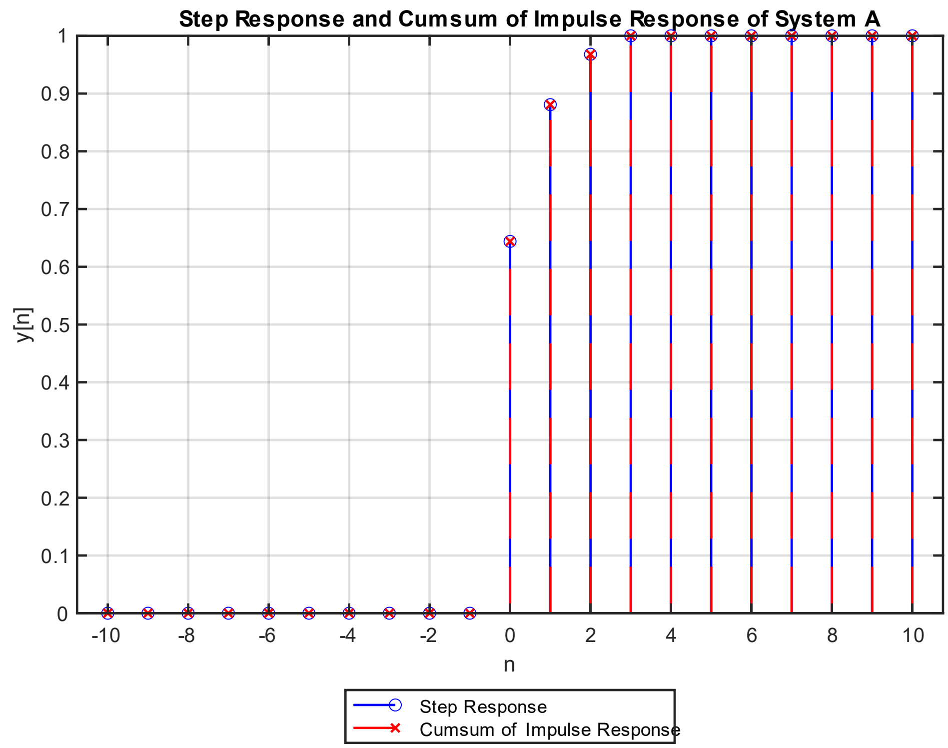
Since the system is given to be LTI, we can further state that:

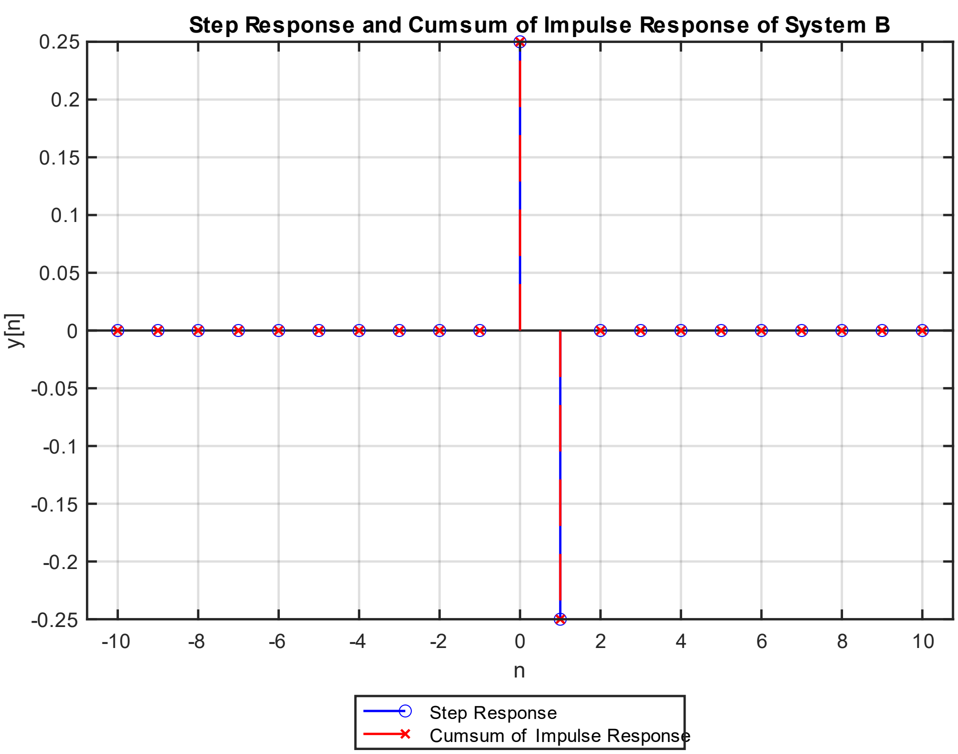
$$s[n] = \sum_{i=-\infty}^{n} h[n]$$

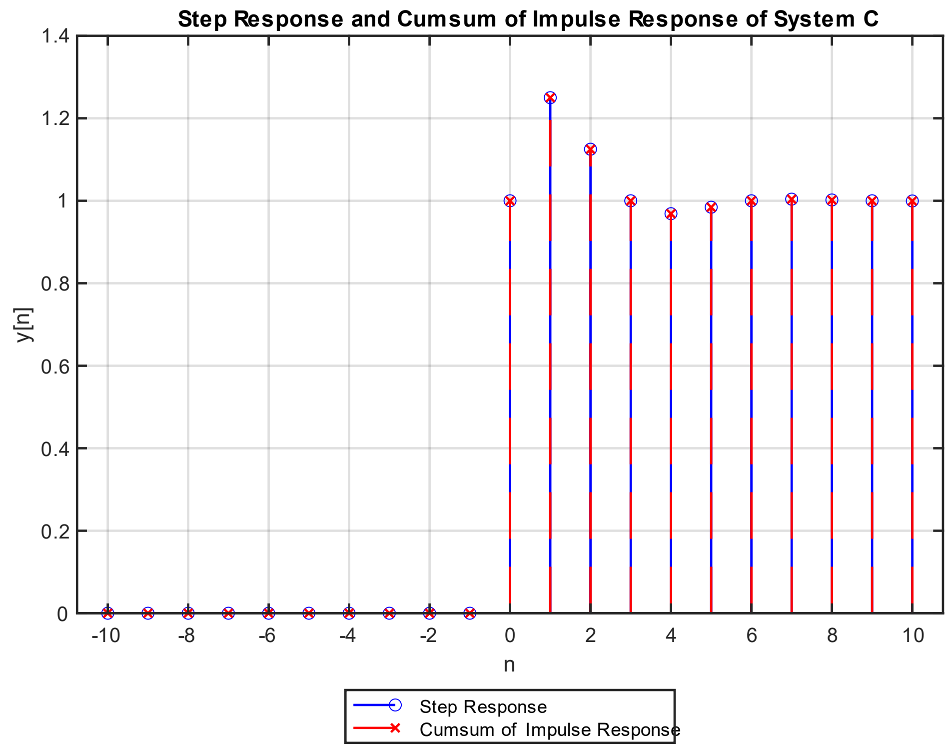
by virtue of the fact that $s[n]$ is the LTI system's output to an input of $u[n]$ and $h[n]$ is the system's output to the input of $\delta[n]$. It should be noted that the summation shown above is computed in the same manner as a cumulative sum and is therefore equivalent. In the developed MATLAB script, the cumulative sum was computed using the $cumsum()$ function.

Informal Graphical Test

Rendering the system outputs for A, B, and C for each the step response and the cumulative sum of the impulse response, the following plots may be obtained. For each, we can see that the two responses are identical and thus it can be concluded that our observations match the aforementioned theory.


Step Response and Cumsum of Impulse Response of System A
y[n]
n
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
Step Response
Cumsum of Impulse Response


Step Response and Cumsum of Impulse Response of System B
y[n]
n
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
-10
-8
-6
-4
-2
0
2
4
6
8
10
Step Response
Cumsum of Impulse Response



Formal Logical Test

The formal logic test for comparing the response of the cumulative sum of the impulse response with the unit step response involved setting a tolerance value, which was chosen to be $1 \cdot 10^{-12}$, and comparing the absolute value of the difference between the two tests with that tolerance. A tolerance is required for the formal logic tests as there is error associated with floating point numbers which can cause direct comparisons between two numbers which should be equal, i.e. value 1 == value 2, to not return true because of residual decimal places in some of the indexes in the arrays.

To accomplish this tolerance comparison the following code is used:

```
ANY(ABS(Y1 - Y2) < TOLERANCE)
```

Running the code with while performing the formal logic test returns **true** for all tests, confirming that the cumulative sum of the impulse response is equal to the unit step response for every LTI system provided.

First Difference

Part IV of the assignment demonstrates that the unit impulse function output $h[n]$ is equivalent to the first difference of the unit step function response $s[n]$. The first difference of a sequence gives the change between consecutive samples and can thus be considered to be the discrete analog of continuous differentiation. It can be stated that:

$$\delta(t) = \frac{d}{dx} u(t)$$

which aligns with the logic utilized in the previous part using the fundamental theorem of calculus. Converting to the analogous discrete formulation of the above:

$$\delta[n] = u[n] - u[n-1]$$

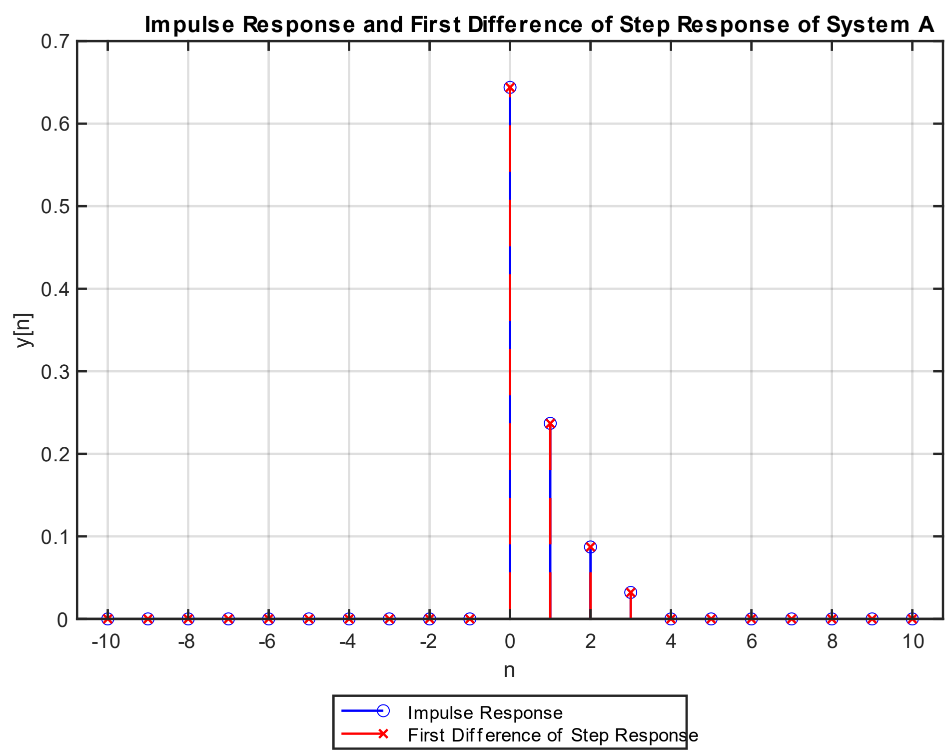
Thus, since the system is LTI, the relationship between the impulse response and the step response can be expressed formally as:

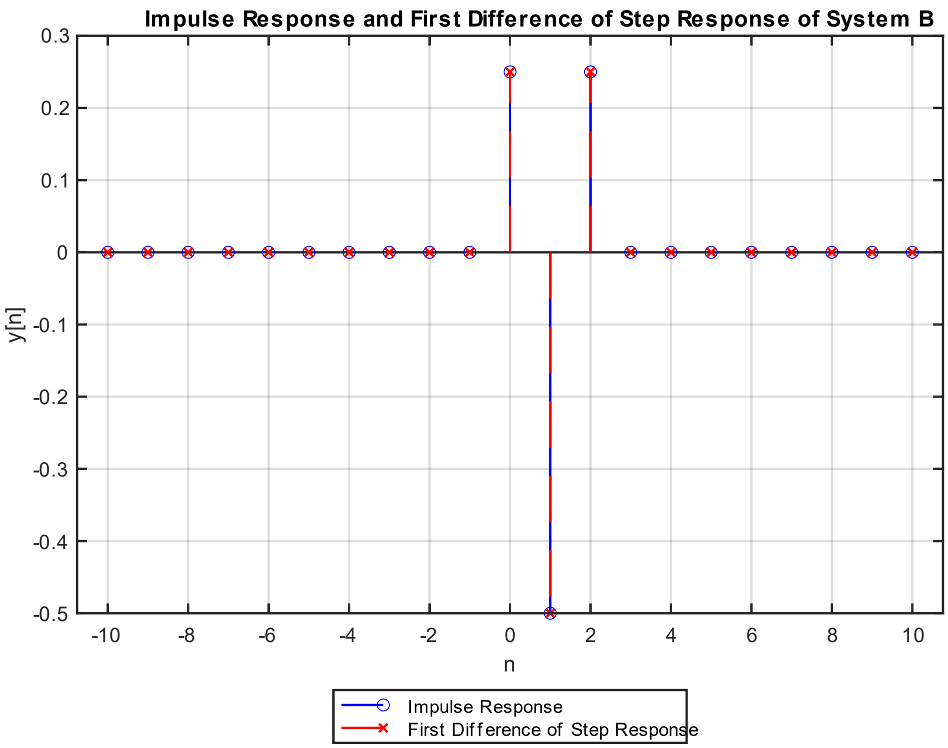
$$h[n] = s[n] - s[n-1]$$

In the developed MATLAB script, the first difference is computed on the vector for the step response s by left-padding the s vector with a 0 to match the dimensions of n and s_diff. As the first difference takes the difference between the ith element and the $i-1$th element, $len(s\_diff) = len(s) - 1$. The left-padding is valid because each system's step response is 0 for $n < 0$. Then, using the $diff()$, the first difference is computed.

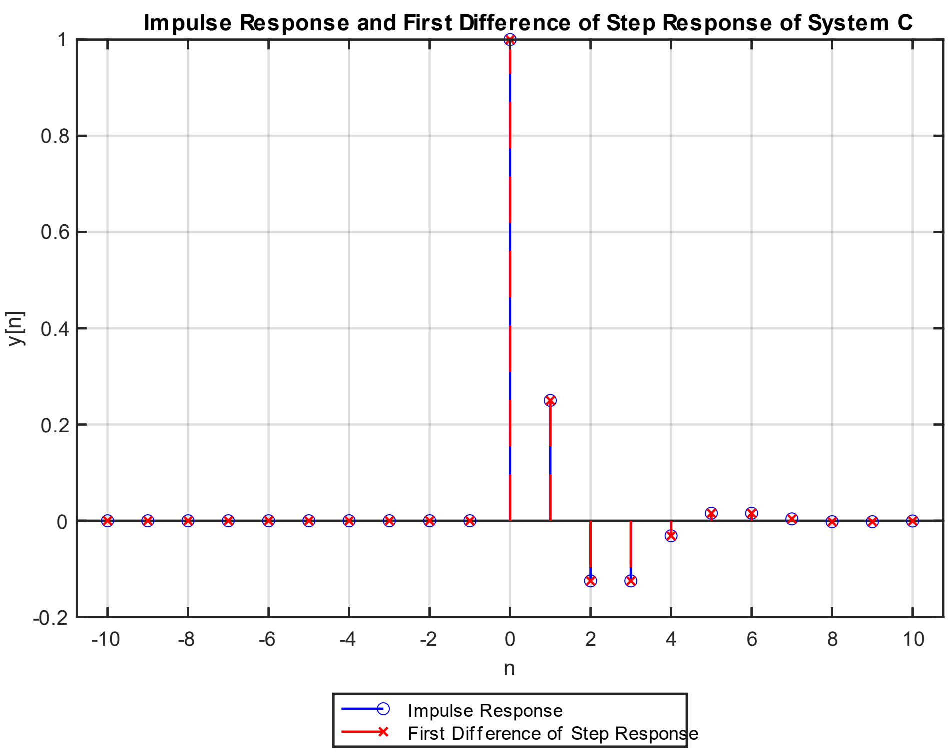
Informal Graphical Test

Rendering the system outputs for A, B, and C for each the impulse response and the first difference step, the following plots may be obtained. For each, we can see that the two responses are identical and thus it can be concluded that our observations match the aforementioned theory.





Impulse Response and First Difference of Step Response of System B
y[n]
n
Impulse Response
First Difference of Step Response



Impulse Response and First Difference of Step Response of System C
y[n]
n
Impulse Response
First Difference of Step Response

Formal Logical Test

The formal logic test for comparing the response of the first difference of the response from the unit step function with the unit impulse response involved setting a tolerance value, which was chosen to be $1 \cdot 10^{-12}$ and comparing the absolute value of difference between the two tests with that tolerance. A tolerance is required for the formal logic tests as there is error associated with floating point numbers which can cause direct comparisons between two numbers which should be equal, i.e. value 1 == value 2, to not return true because of residual decimal places in some of the indexes in the arrays.

To accomplish this tolerance comparison the following code is used:

```
ANY(ABS(Y1 - Y2) < TOLERANCE)
```

Running the code with while performing the formal logic test returns **true** for all tests, confirming that the first difference of the unit step response is equal to the unit impulse response for every LTI system provided.

Real Biomedical Signals

Parts V, VI, & VII move away from generic input signals like the unit step function and begin to explore these systems and the convolution property of LTI systems with real biomedical signals. Specifically, an ECG and respiration signal are each tested. Depending on the system, it may not be ideal to directly compute the system response for every input signal. This could be because the system takes a long time to calculate the output, it is computationally expensive, or there may not always be access to the system. In these cases, it would be useful to have another way to determine the output signal of a system with a given input without having to directly compute it. Luckily there is another way, if it is an LTI system, the convolution of the impulse response with the input signal will equal the system's output signal. This allows the system's output signal to be determined without the need to pass the input signal through the system avoiding the possible disadvantages discussed.

The convolution is a mathematical operation that in a sense describes how two signals pass through each other. There are two notations for the convolution operations and conveniently the simpler form shown below is the same for both continuous and discrete time.

$$y(t) = h(t) * x(t)$$

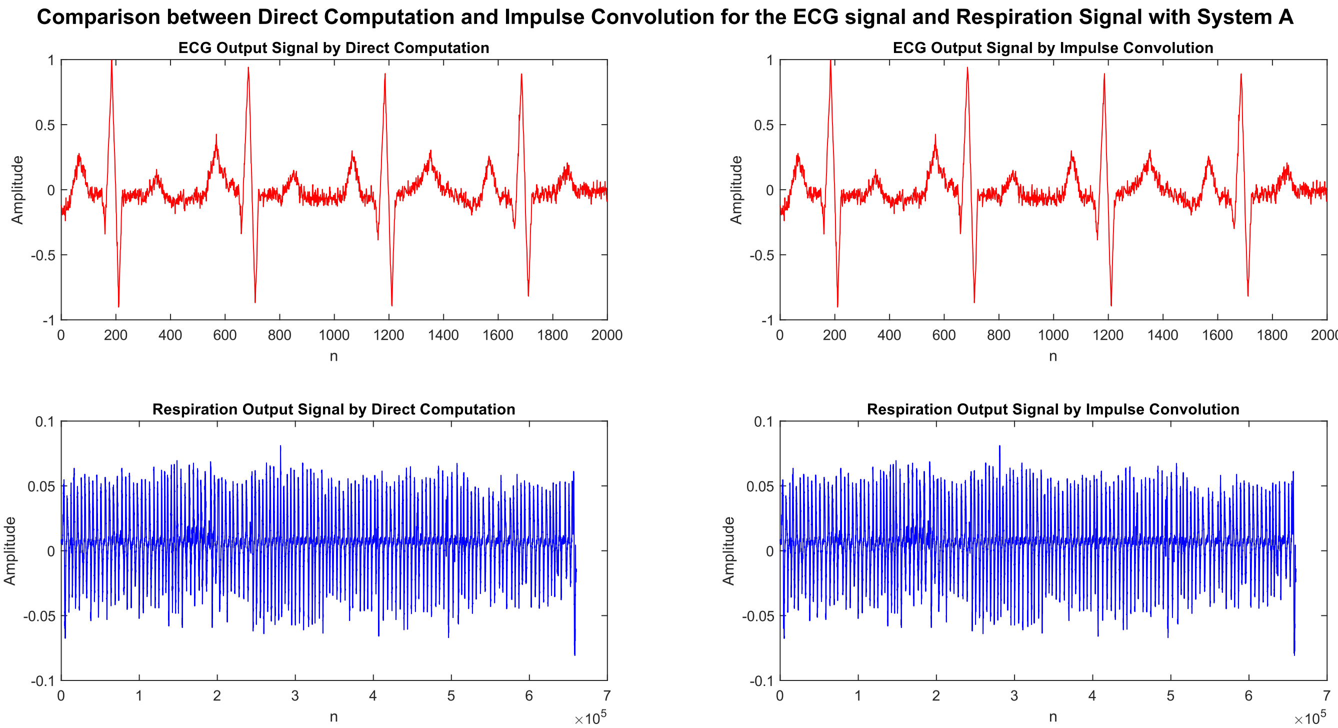
$$y[n] = h[n] * x[n]$$

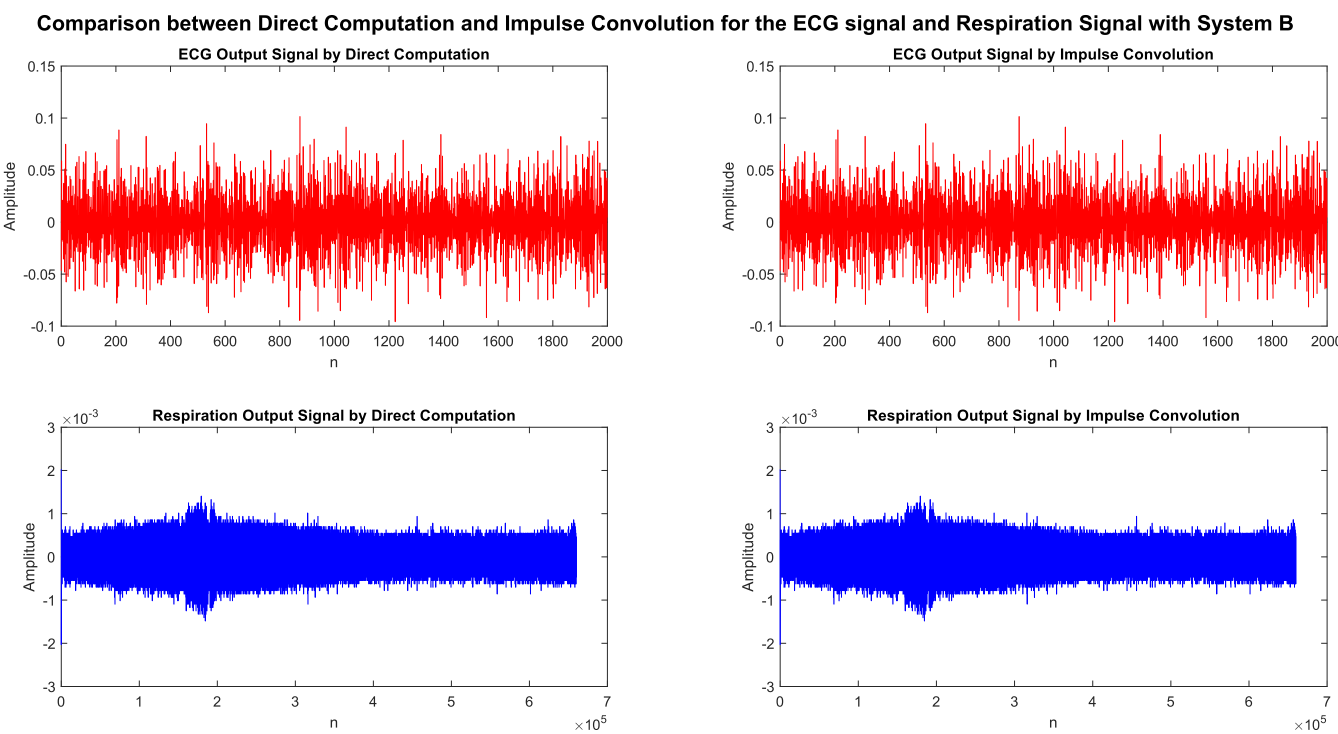
Specifically, this part of the assignment intends to test whether this statement is true, that the output signal of a discrete LTI system is equal to the convolution of the unit impulse response with the input signal.

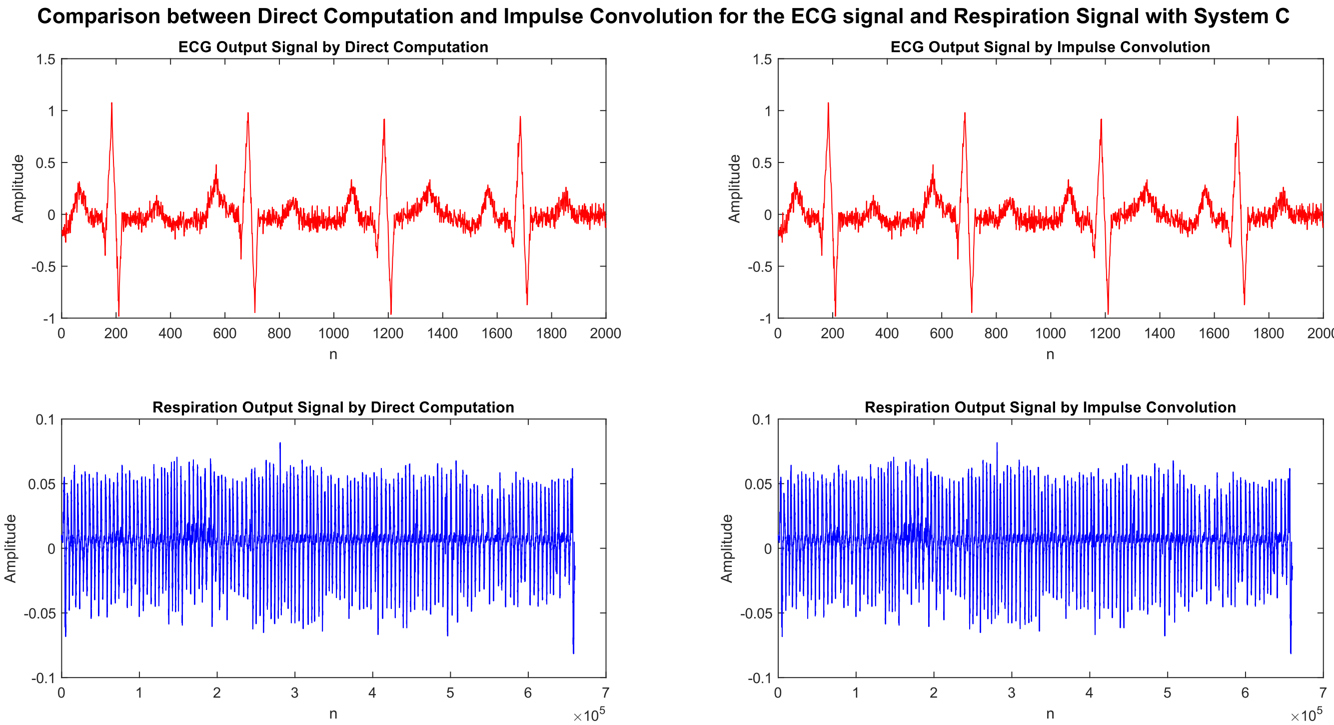
In the developed MATLAB script, the ECG file is loaded to get the ECG input signal. The system response is directly computed for both the ECG input signal and unit impulse. The ECG input signal and unit impulse response are then convolved. The two output signals are then truncated to the same signal length to ensure effective comparison. This comparison is completed with a graphical and logical test both shown below. This same process is then repeated with the respiration input signal.

Informal Graphical Test

Rendering the system outputs for A, B, and C for each input signal gives the directly computed output signal and the output signal computed through convolution, generating the following plots. For each, we can see that the two output signals are identical and thus it can be concluded that our observations match the aforementioned theory for all three provided LTI systems.


Comparison between Direct Computation and Impulse Convolution for the ECG signal and Respiration Signal with System A
ECG Output Signal by Direct Computation
ECG Output Signal by Impulse Convolution
Respiration Output Signal by Direct Computation
Respiration Output Signal by Impulse Convolution
Amplitude
n
×10^5


Comparison between Direct Computation and Impulse Convolution for the ECG signal and Respiration Signal with System B
ECG Output Signal by Direct Computation
ECG Output Signal by Impulse Convolution
Respiration Output Signal by Direct Computation
Respiration Output Signal by Impulse Convolution
Amplitude
n
×10^-3
×10^5



Formal Logical Test

When the MATLAB script is run, the formal logical test will print to the terminal that the "convolution with impulse response and direct computation are equivalent for System *X*" as long as both the ECG and respiration signals pass the test. The test is completed by the following line of code:

```
ANY(ABS(Y1 – Y2) < TOLERANCE)
```

This functions by first subtracting each element of one of the output signals from each element of the other output signal. If they are identical this will create a vector of zeros or near zeros accounting for possible floating-point arithmetic errors. To ensure these near zeros which only result from floating-point arithmetic errors do not cause a false test failure the absolute value of each is taken and compared to a tolerance. The "any" function compares each element to the tolerance to ensure they are all less than the tolerance otherwise the test fails, and the following message will be output to the terminal "convolution with impulse response and direct computation are NOT equivalent for System *X*." All three provided systems pass this formal logical test and match the aforementioned theory.

Bonus 1

Bonus one was attempted for this assignment. Every time the MATLAB script is run, there will be a terminal output stating whether each system passes or fails each logical test. Please note that the logical tests are not explained in this section but rather explained in the section of this report about that LTI system property. Each logical test passes for each property and each system agreeing with the graphical results provided in the previous sections.

Bonus 2

Bonus two was attempted for this assignment. Bonus two required the analysis of all three systems to determine if they were a high-pass, low-pass, or band-pass filter. To accomplish this the DFT (Discrete Fourier Transform) of every system was taken over a large range of points to transform it into the frequency spectrum. The equation for the DFT is the following:

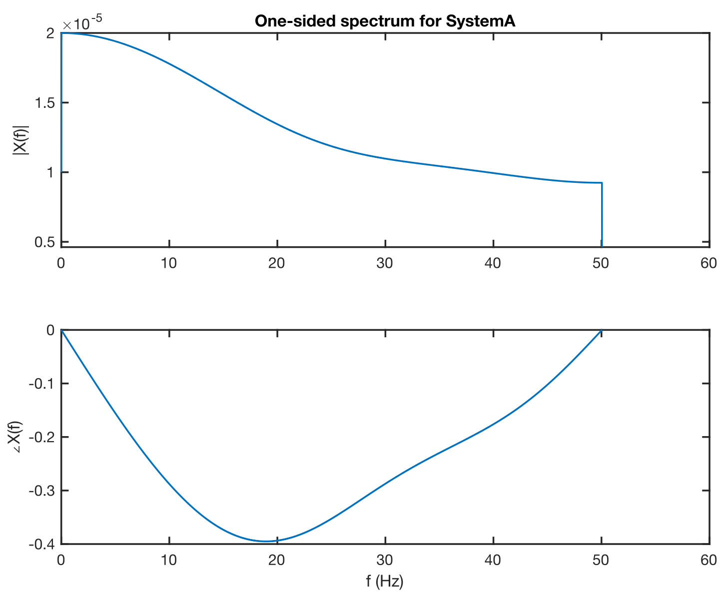
$$X_k = \sum_{n=0}^{N-1} x[n]e^{-\frac{j2\pi kn}{N}}, k = 0,1,\dots,N-1$$

Where N is the number of time steps. Taking the DFT of the unit impulse response of the system is useful because of the property that convolution in the time domain is equivalent to multiplication in the frequency domain. As a result, the previous relationship discussed that for LTI systems the output signal can be computed by taking the convolution between the unit impulse response and the input signal would become multiplication of the input signal's DFT and unit impulse response's DFT in the frequency domain. If the magnitude of the unit impulse response DFT is lower at a given frequency that would mean that the system would filter out any of those frequencies from a given input signal. This is the logic used to test the filtering properties of each system.

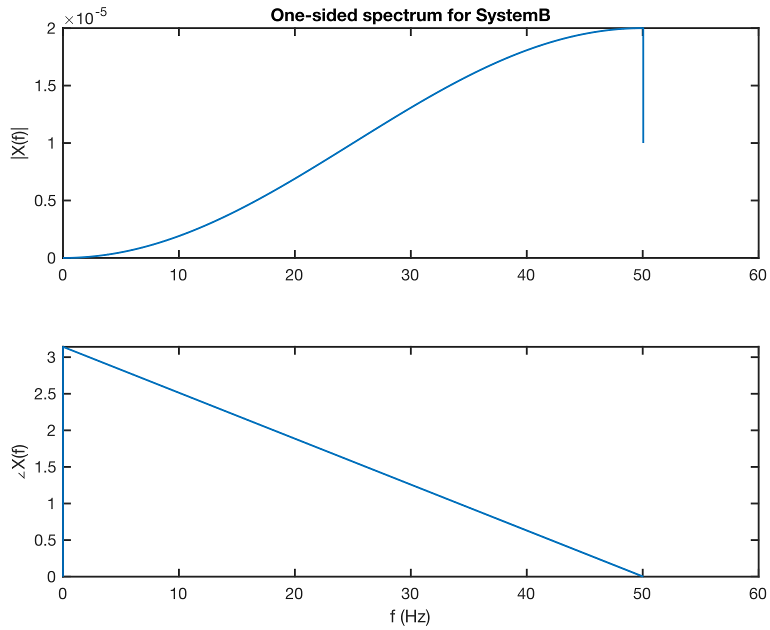
The tests were run at multiple different sampling frequencies, as the LTI system files were able to interpolate time steps where $n \in \mathbb{Q}$. This allowed us to choose n values from 0 to 999 and sample 100,000 different values, giving a sampling frequency of $\approx 100\ Hz$. The goal with checking different sampling frequencies was to analyze the effects of aliasing on the system as we do not know what the Nyquist frequency of this system is.

Checking the system with larger sampling frequencies only resulted in a smoother curve for the $|X_k|$ vs f graph.

Below is the frequency spectrum for System A. It is clear from the large magnitude of frequencies at the low end of the frequency spectrum that the system is acting as a **low-pass filter**.



Below is the frequency spectrum for System B. It is clear from the large magnitude of frequencies at the high end of the frequency spectrum that the system is acting as a **high-pass filter**.



Finally, below is the frequency spectrum for System C. The system has a band of frequencies that it allows through near the lower end of the frequency spectrum. This implies that the system is acting as a **band-pass filter**.