Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title	Control Systems
Course Number	ELE632
Semester/Year	W25

Instructor	Dr. Dimitri Androutsos

Lab #	2
Lab Title	Time-Domain Analysis of Discrete-Time Systems (Part 2)

Submission Date	February 16, 2025
Due Date	February 16, 2025

Last Name	**First Name**	**Student Number**	**Section**	**Signature**
Kandasamy	Heitan	501184804	04	HK

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.pdf

A Unit Impulse Response

A.1 Finding Impulse Response h[n] using python lfilter

A.1.1

$$y[n] + \frac{1}{(4+1)}y[n-1] - \frac{1}{(4+1)}y[n-2] = \frac{1}{(8+1)}x[n] \tag{1}$$

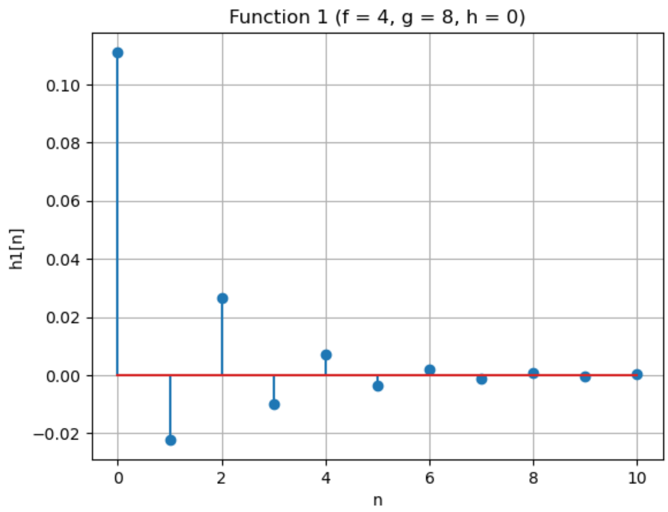

Figure 1: Impulse Response of Function 1

A.1.2

$$y[n] + \frac{1}{(0+1)}y[n-2] = x[n] \tag{2}$$

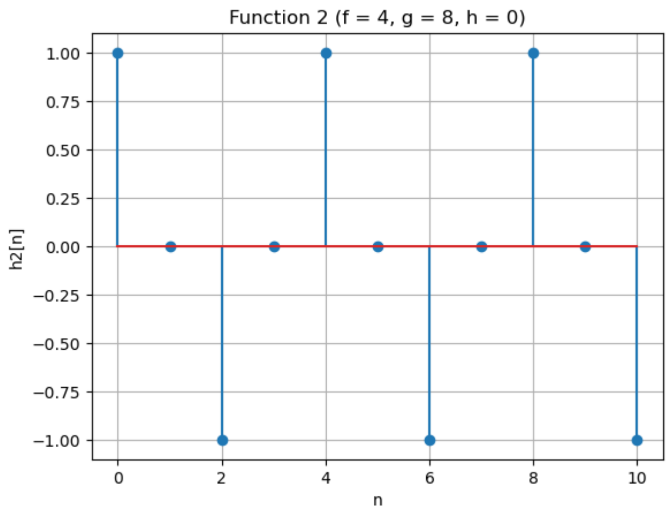

Figure 2: Impulse Response of Function 1

A.2 Calculating h[n] By Hand and Calculation of h[3]

A.2.1

$$y[n] + \frac{1}{(4+1)}y[n-1] - \frac{1}{(4+1)}y[n-2] = \frac{1}{(8+1)}x[n] \tag{3}$$

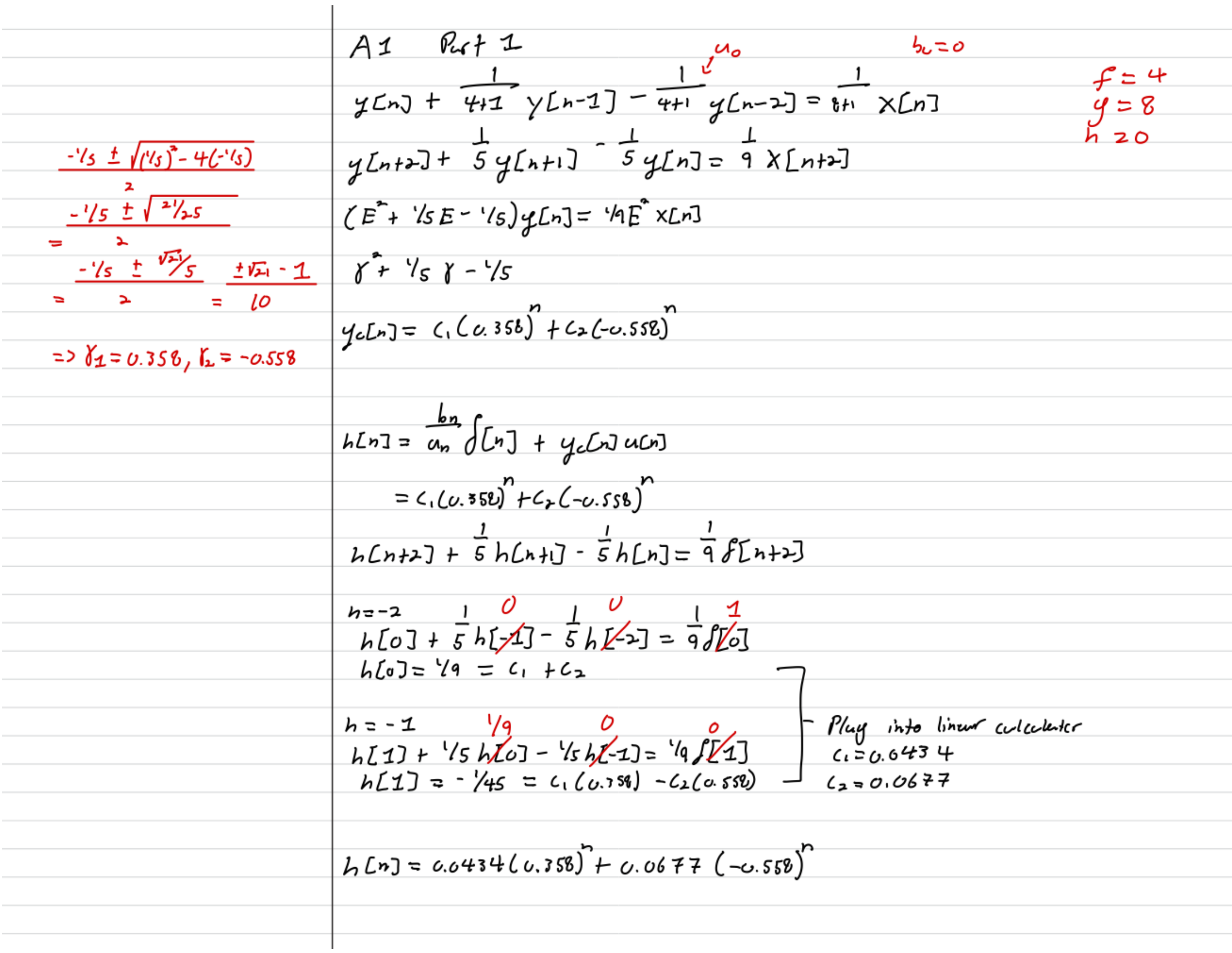

Figure 3: Hand Calculation of Impulse Response for Function 1

General solution

$$h[n] = (4.34 * 10 - 2(0.358)n + 6.77 * 10 - 2(-0.558)n) * u[n] \tag{4}$$

Solving for h[3]

$$\begin{aligned} h[3] &= 4.34 * 10^{-}2(0.358)^3 + 6.77 * 10^{-}2(-0.558)^3 \\ &= -9.77 * 10^{-}3 \end{aligned}$$

In this case, the calculations done in *Figure 3* and the generated impulse response from python match, as both values for h[3] are equal to -0.00977.

A.2.2

$$y[n] + \frac{1}{(0+1)} y[n-2] = x[n] \tag{5}$$

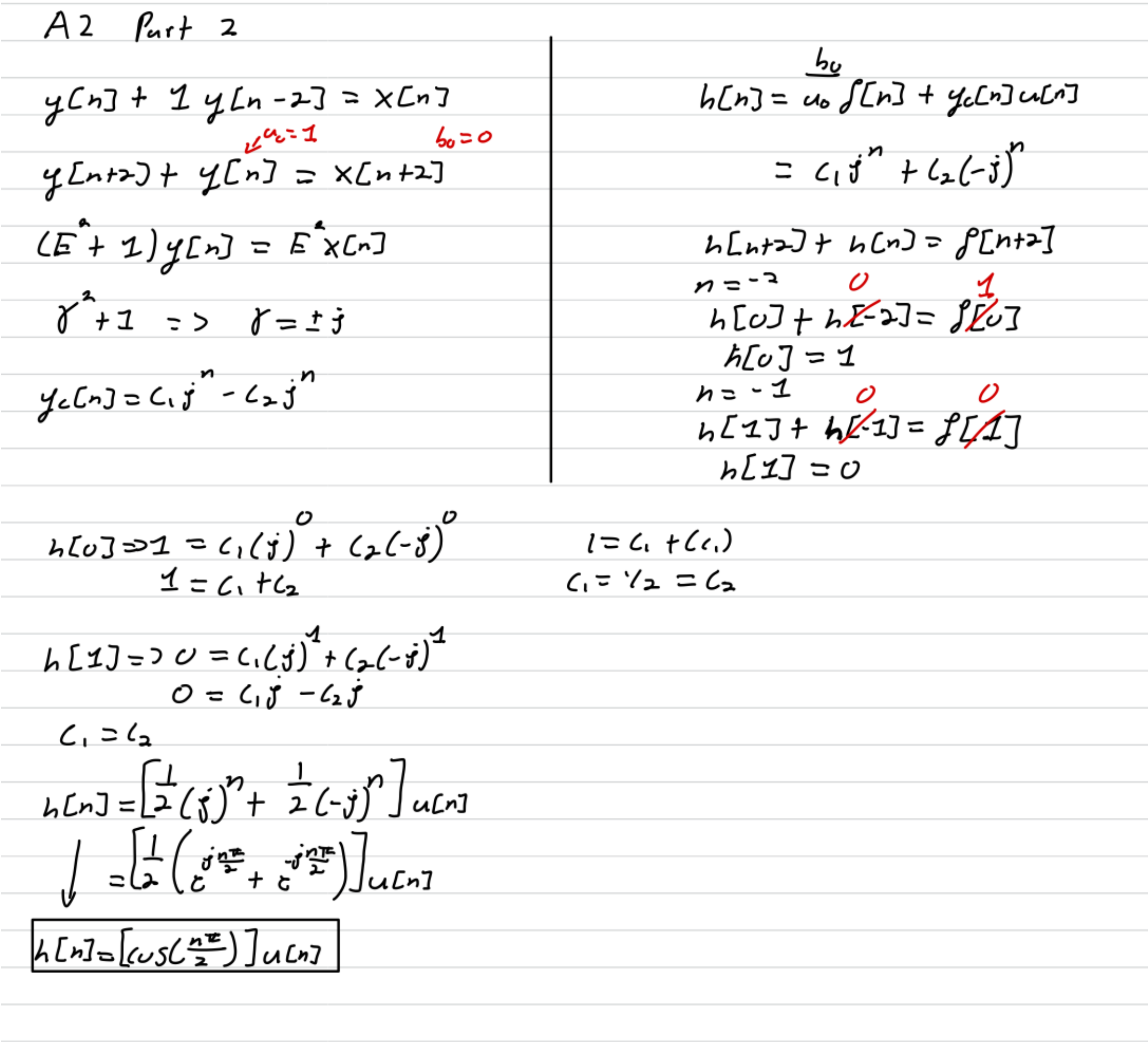

Figure 4: Hand Calculation of Impulse Response for Function 2

General solution

$$h[n] = cos(\frac{n\pi}{2}) * u[n] \tag{6}$$

Solving for h[3]

$$\begin{aligned} h[3] &= cos(\frac{3\pi}{2}) \\ &= 0 \end{aligned}$$

The calculation done bny Python's lfilter library and the calculations by hand both match where h[3] is equal to 0. This verifies that the library is working as intended.

B Zero Input Response

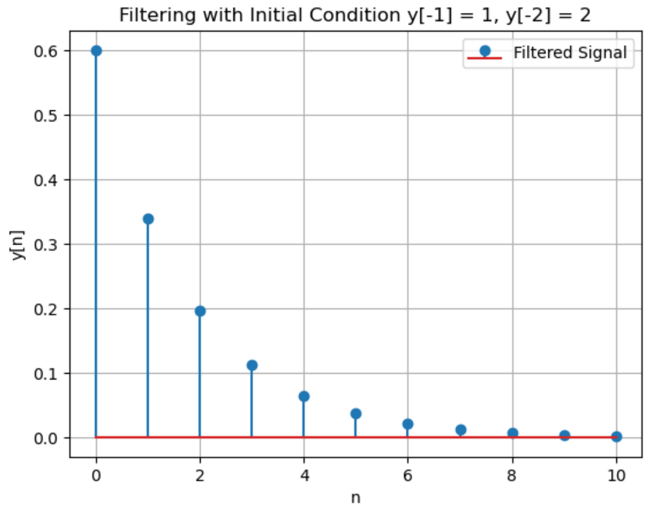

Figure 5: Zero-Input Response of System as Defined in Part B

This Graph shows the zero-input response of the following system *(Equation 7)* with initial conditions of y[-1] = 1, y[-2] = 2

$$y[n] - \frac{4}{10}y[n-1] - \frac{1}{10}y[n-2] = 2x[n] \tag{7}$$

C Zero-State Response

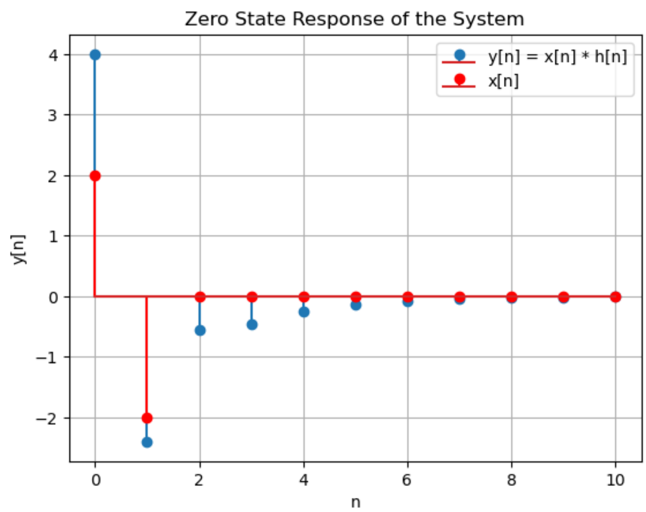

Figure 6: Zero-Input Response of System as Defined in Part B

This Graph shows the zero-State response of the system in Part B *(Equation 7)* with *Equation 8* as the input

$$x[n] = 2cos(\frac{2\pi n}{2})(u[n] - u[n-2]) \tag{8}$$

D Total response

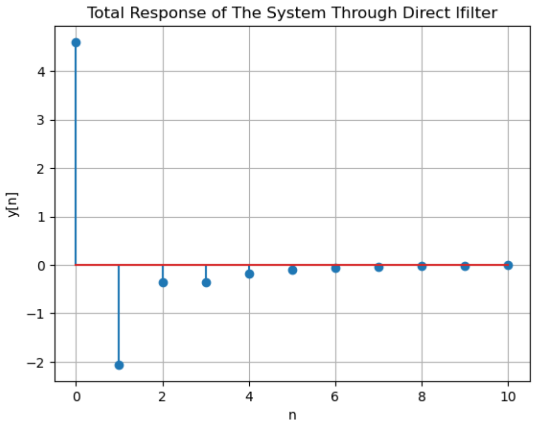

Figure 7: Total Response Through Python lfilter

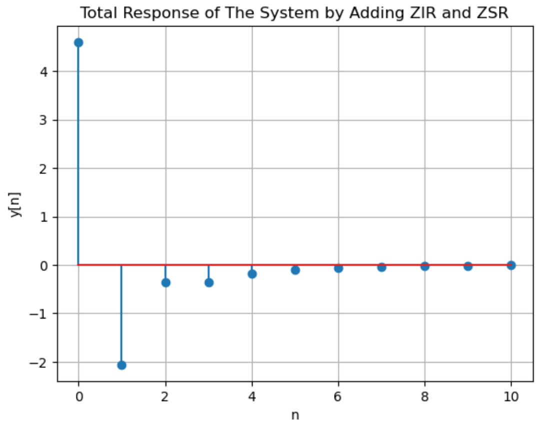

Figure 8: Total Response Through Summation of Zero-Input Response and Zero-State Response

Given that both Figure 7 and Figure 8 are the same, this proves that it is possible to find the total response of a system by taking the summation of the zero-input and zero-state response of the system.

E Convolution and System Stability

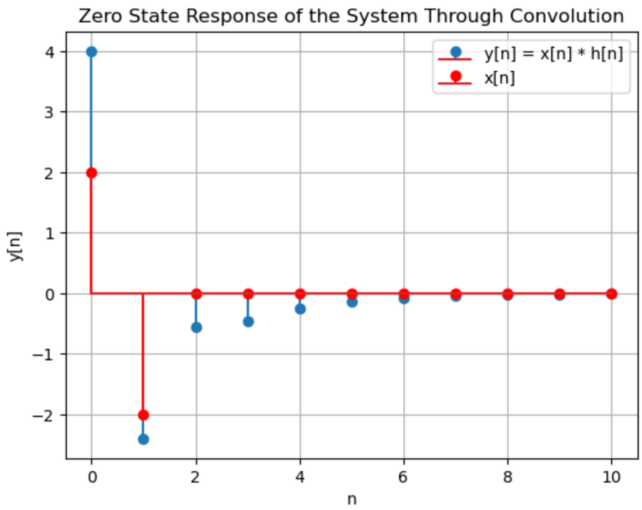

Figure 9: Zero-State response of System Described in Part B With Input in Part C

This output is exactly the same as seen in Part C (Figure 6). The zero-state response is the same as taking the convolution between the system's impulse response and the input.

This system is considered asymptotically stable as the response can be absolutely summable. The values do not grow infinitely, but rather converge at zero. It can also be shown mathematically as the roots of the characteristic equation have an absolute value less than 1.

$$y[n] - \frac{4}{10}y[n-1] - \frac{1}{10}y[n-2] = 2x[n]$$
$$(E^2 - \frac{4}{10}E - \frac{1}{10})y[n] = 0$$
$$\gamma^2 - \frac{4}{10}\gamma - \frac{1}{10} = 0$$
$$Apply\ Quadratic\ Formula$$
$$\gamma_1 = 0.574, \gamma_2 = -0.174$$

Both of which are between -1 and 1, resulting in internal stability.

F Moving Average Filter

given the following System:

$$y[n] = \frac{1}{N}(x[n] + x[n-1] + ... + x[n-(N-1)]) \tag{9}$$

We want to design a differential equation that has an impulse response $h[n] = \frac{(u[n]-u[n-N])}{N}$.

By analyzing the system, it is determined that it is a finite impulse response (FIR). This means that there should be no characteristic equation, and the system is made up of only impulses, the feed-forward coefficients. Based on this, the constant coefficient difference equation that has the aforementioned impulse response would simply be:

$$y[n] = \frac{1}{N}x[n] + \frac{1}{N}x[n-1] + ... + \frac{1}{N}x[n-(N-1)] \tag{10}$$

For whatever value N is, that is how many impulses there will be, with a coefficient of $\frac{1}{N}$.

By simply creating a Python function that returns a list of N terms, each with a value of $\frac{1}{N}$, and a list of one term with a value of 1, the feed-forward and feedback coefficients can be determined for the system.

```
def movingAvg(N):
    # System is FIR, Therefore a = 1
    a = [1] # FeedBack coefficients

    # all coefficients are 1/N
    b = np.ones(N)/N # Feed forward coefficients
    return b, a
```

By testing the filter with the input of length-45 below:

$$x[n] = cos(\frac{\pi n}{2}) - \delta[n-9] + \delta[n-1] \tag{11}$$

The Plots can be observed in *Figures 10, 11, and 12.*

These plots show that as the moving-average filter increases in N, the values are less spread out, essentially smoothing out the signal. There are still some sudden spikes in the values, but this is mainly due to the δ functions being spread out at [n - 9] and [n - 1].

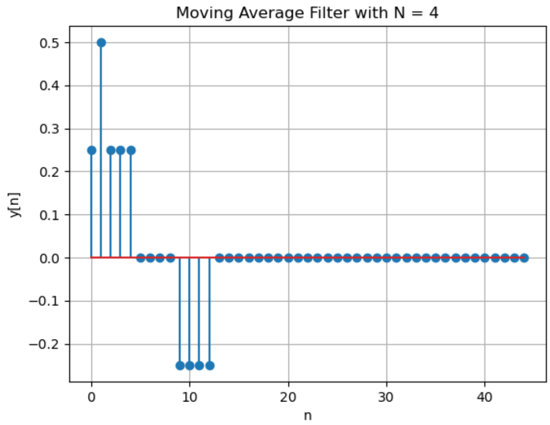

Figure 10: Moving Average Filter with N = 4

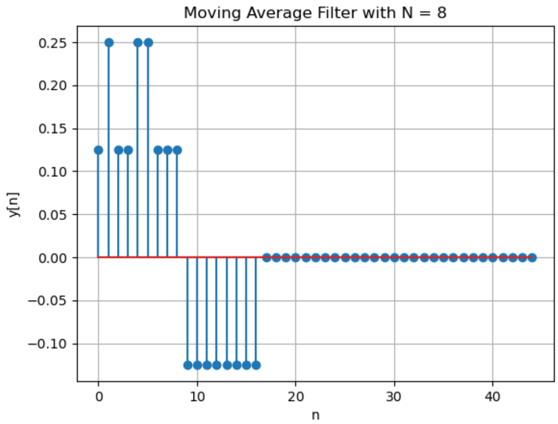

Figure 11: Moving Average Filter with N = 8

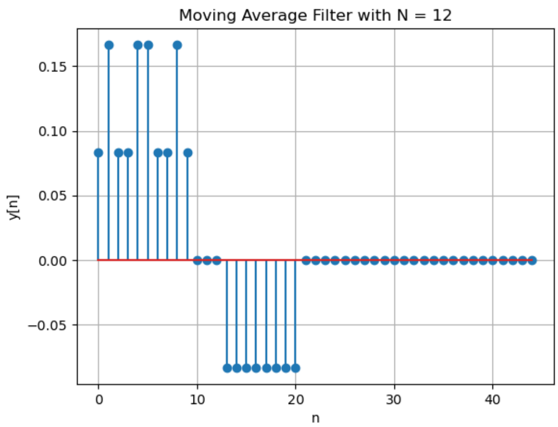

Figure 12: Moving Average Filter with N = 12