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

Toronto Metropolitan University

Department of Electrical & Computer Engineering

BME 639 - Control Systems & Bio-Robotics
Lab Report

Semester/Year	W2024
Lab Number:	1

Instructor:	Dr. Owais Khan
Section:	02
Submission Date:	February 6 2024
Due Date:	February 6 2024

Student Name (ID):	Mathew Szymanowski (501094808)
Signature:	
Student Name (ID):	Tanvir Hassan (501104056)
Signature:	

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: www.ryerson.ca/senate/current/pol60.pdf. Before submitting your report, your TA asks questions about your report. If there is no consistency between your oral answer and your report, you will lose 50% of your total mark. For this part, Pass or fail will be circled next to your name accordingly

PART 1

A: Introduction to Simulink

A.1

Creating a model in the Simulink toolbox.

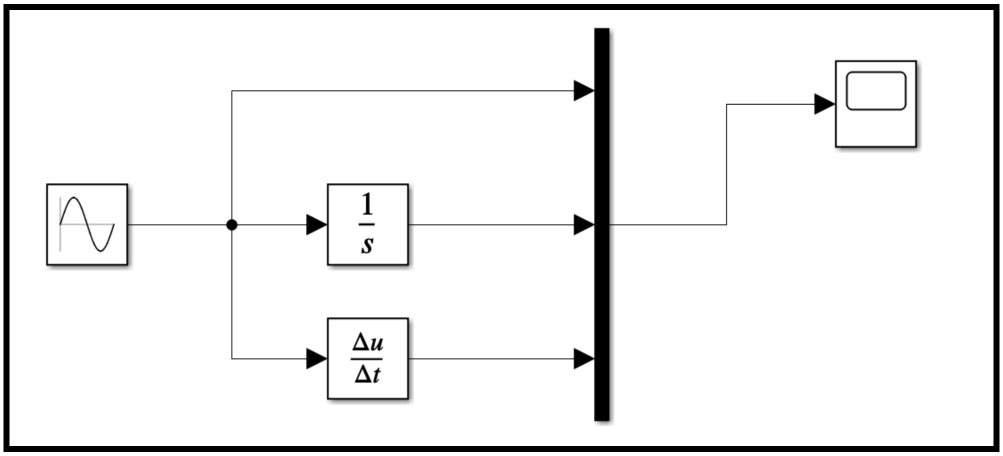


Fig 1: Simulink Model

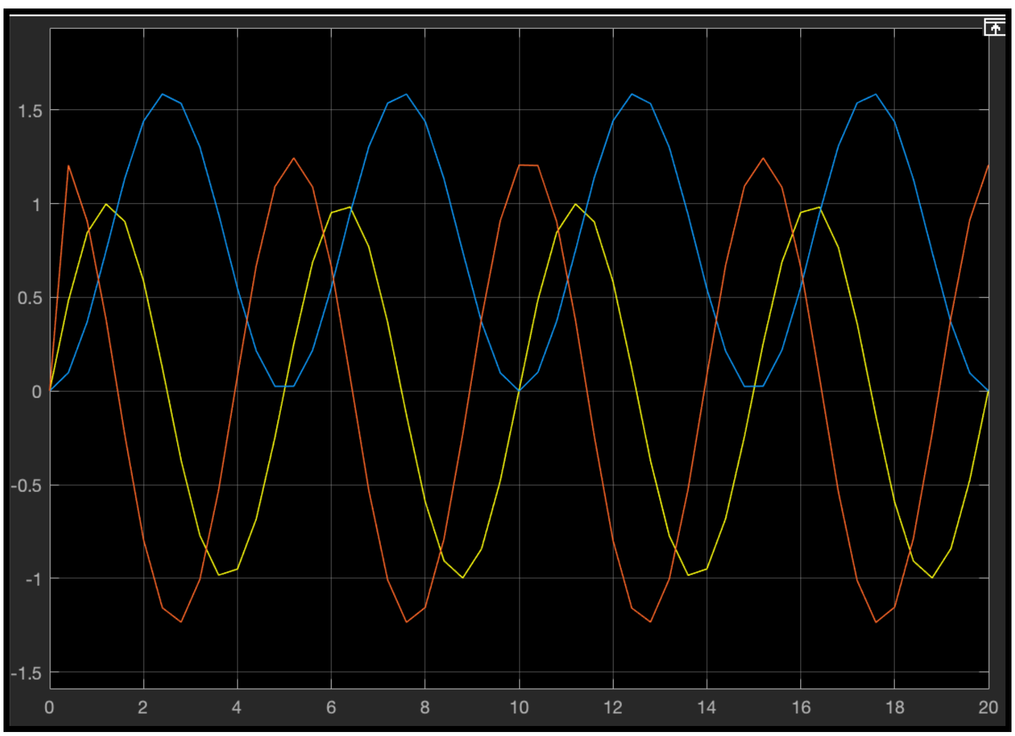


Fig 2: Scope output from the Simulink model

A.2

The out.simout block was used to send an array of data to Matlab where it can be plotted to compare the outputs.

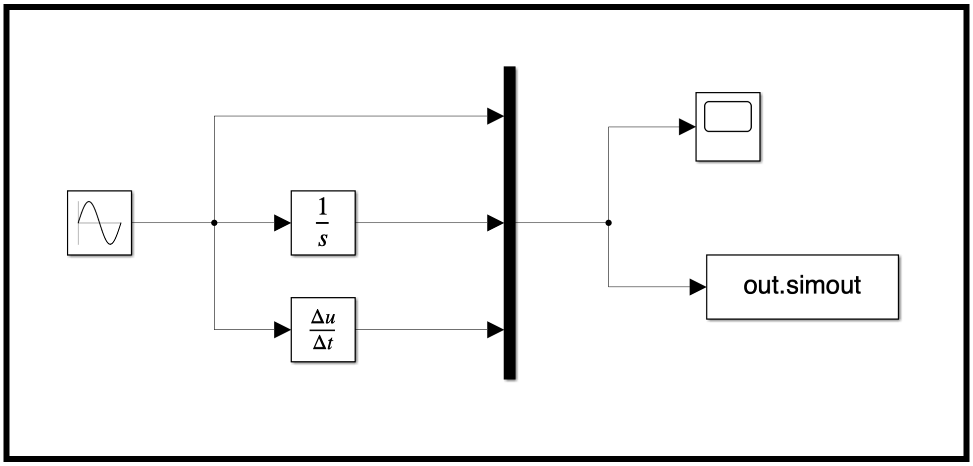


Fig 3: Simulink Model

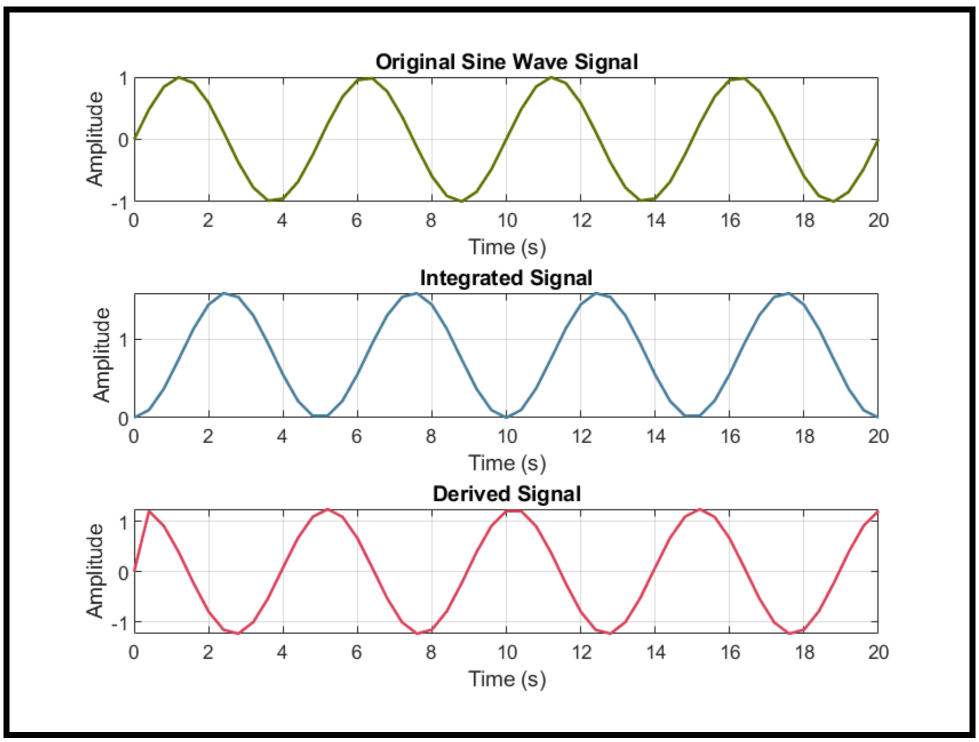


Fig 4: Sine wave, its integral and its derivative

A.3

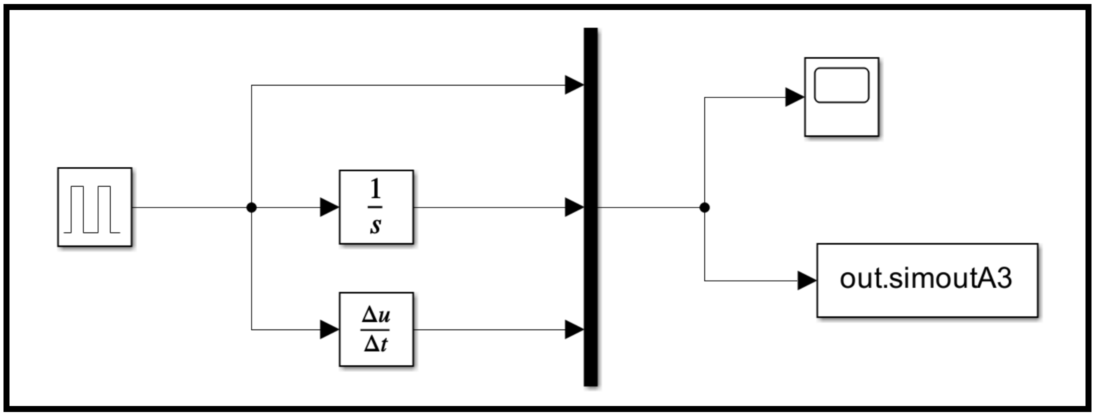


Fig 5: Simulink Model of a pulse function

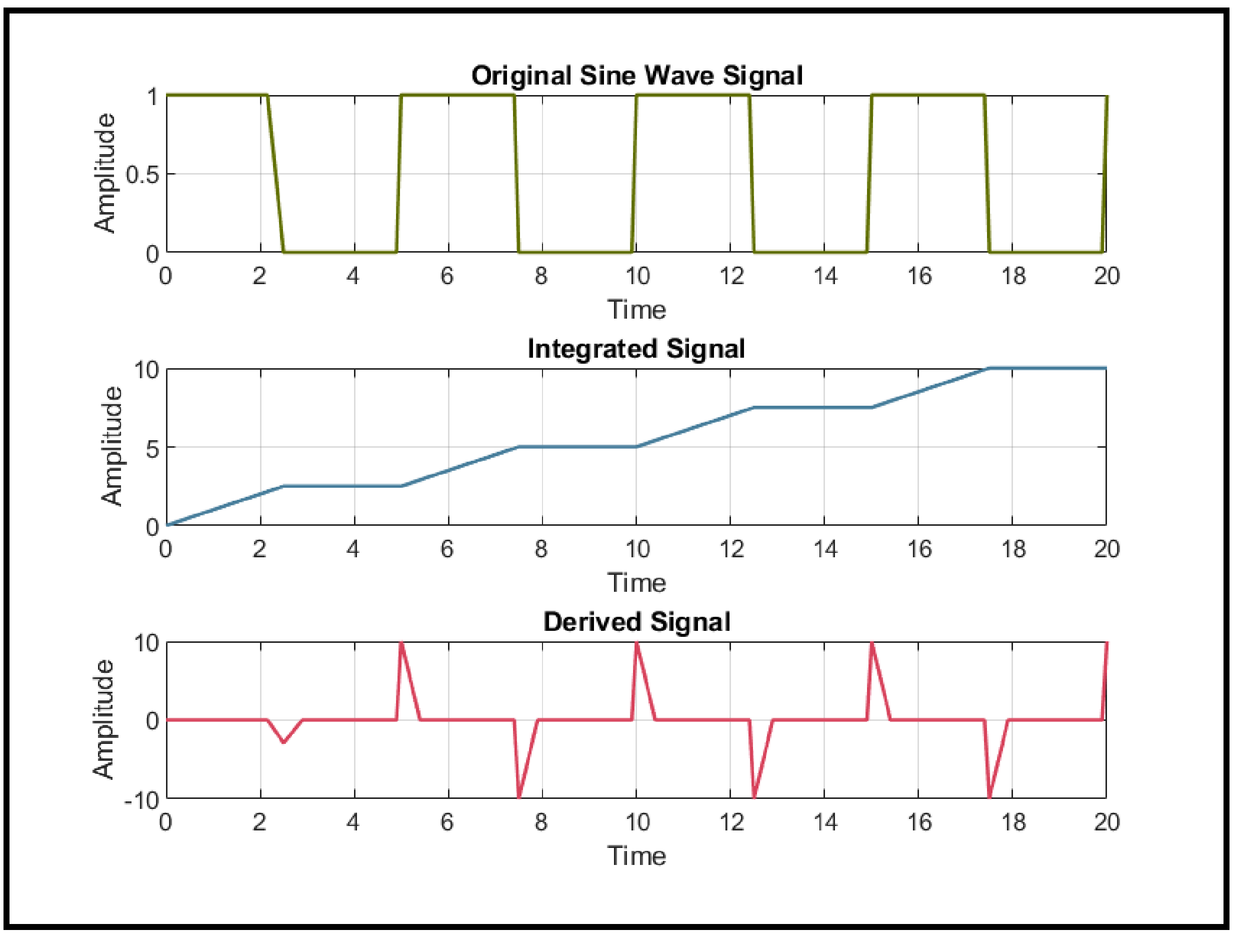


Fig 6: Pulse function, its integral and its derivative

A.4

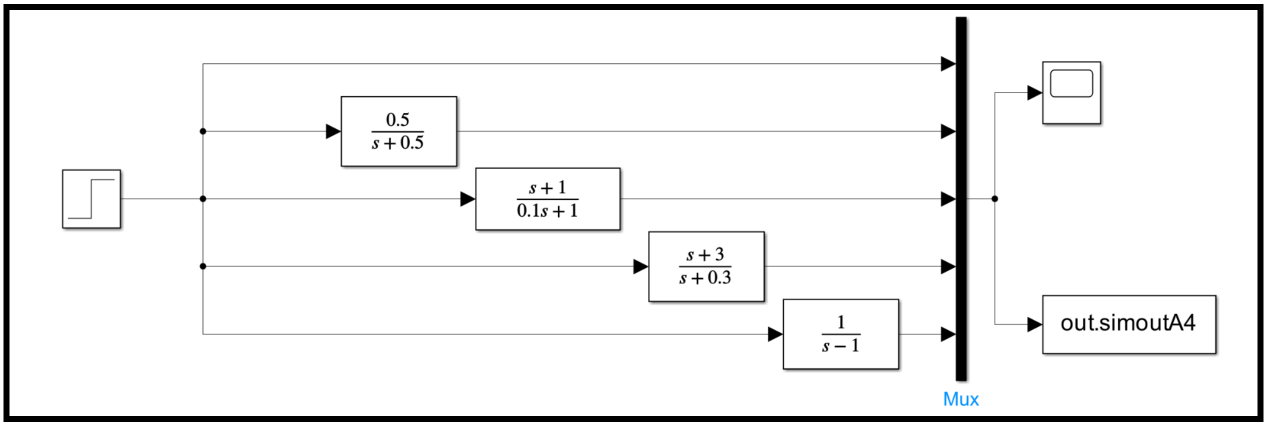


Fig 7: Simulink Model of Unit step and transfer function

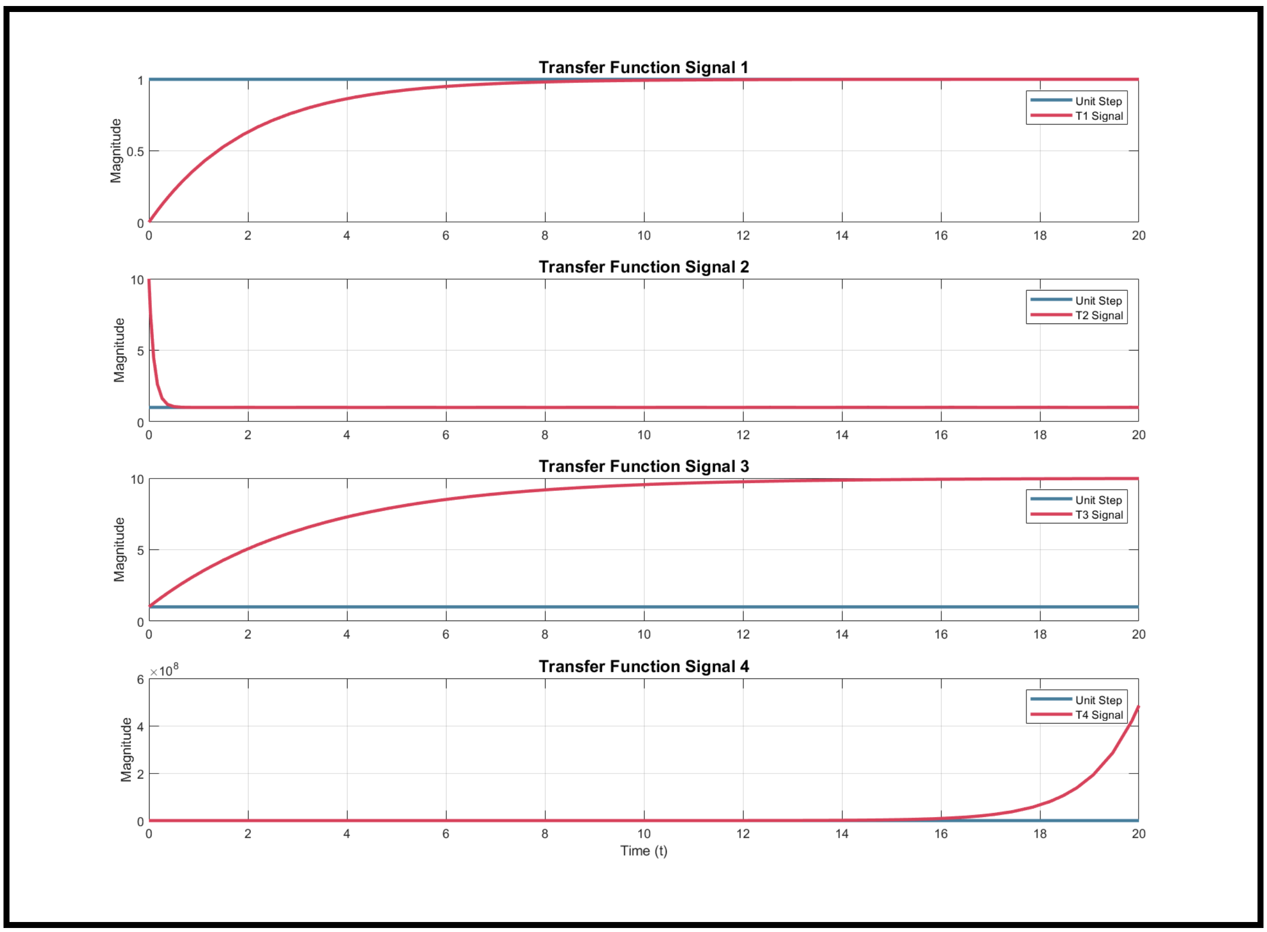


Fig 8: Response of transfer function to Unit Step Signal

B:Time Response of First-Order Systems

B.1

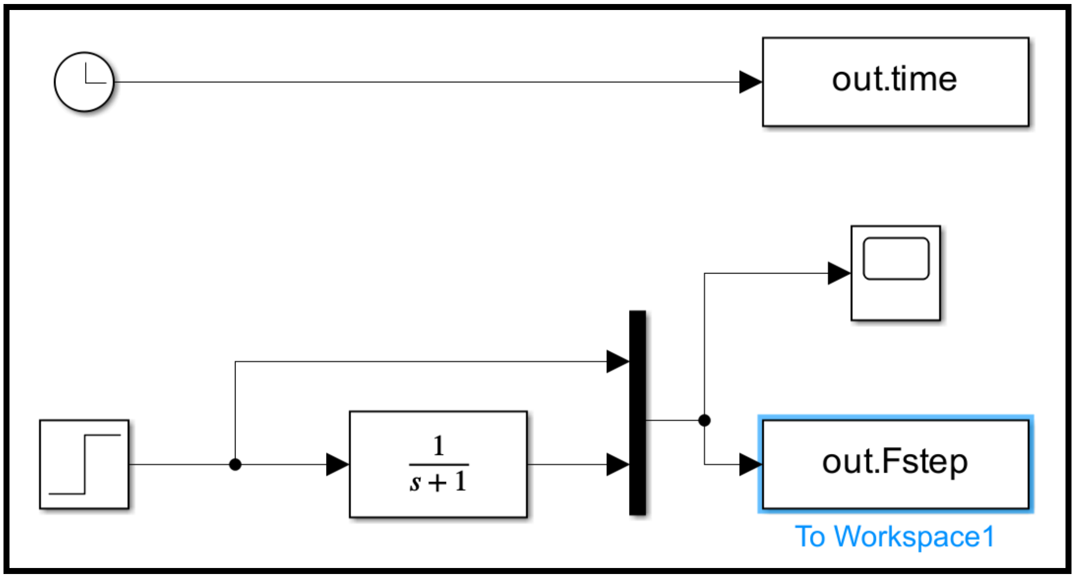


Figure 9: Print of Simulink Model

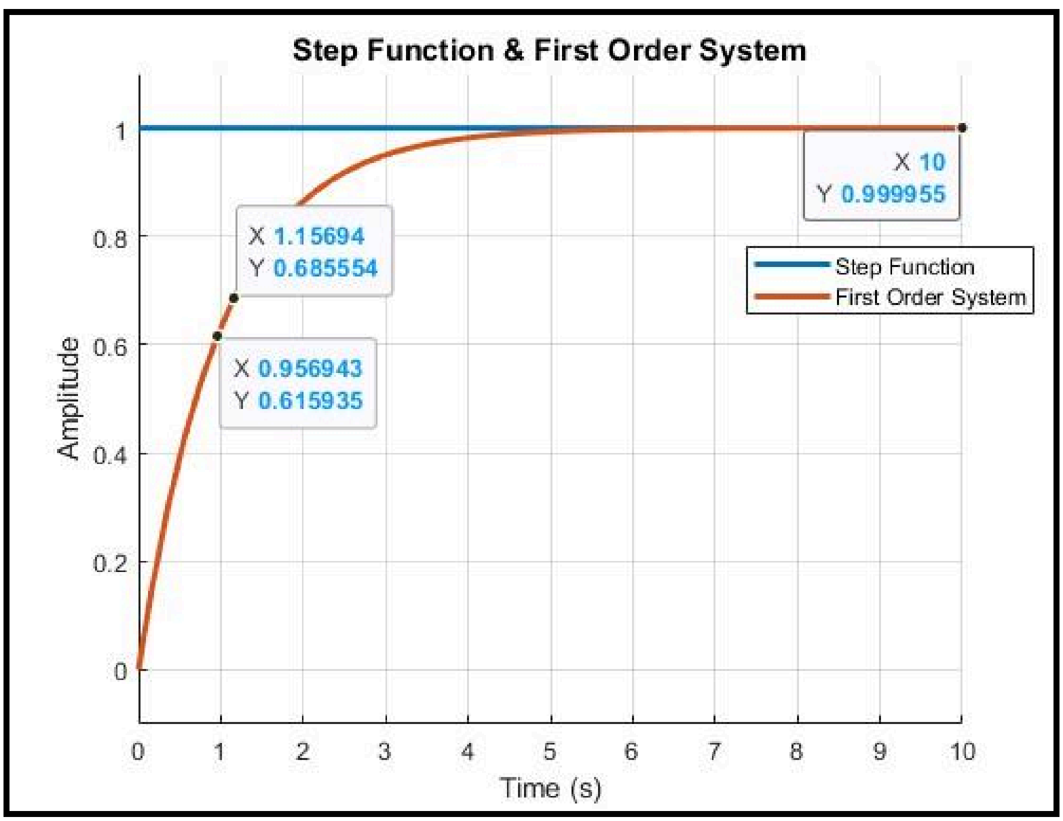


Figure 10: Plotted Input and Step Response of System

The measured gain as shown in the graph above at steady state (10 seconds) rounds to 1. The time constant should be 63% of the output. Using a linear interpolation between the two available data points the measured time constant was 1 as is seen in figure 11.

$$y = y_1 + (x - x_1)\frac{(y_2 - y_1)}{(x_2 - x_1)} \qquad x_1 = 0.9569 \qquad y_1 = 0.6159$$

$$x_2 = 1.1569 \qquad y_2 = 0.6855$$

$$x = 0.9974$$

$$\therefore \text{measured } \tau = 0.9974 \simeq 1 \qquad y = 0.63$$

Figure 11: Linear Interpolation Calculation of the Time Constant

Comparing to the theoretical values, both the gain and time constant, when rounded are the same. When the system is arranged in the form shown in Figure 12, the theoretical gain is denoted as k and the theoretical time constant is denoted as τ. Given that the system is already in that form as shown in Figure 9, both, the gain and time constant, should be 1.

$$\frac{k}{\tau s+1}$$

Figure 12: First Order System with Gain (k) and Time Constant (τ) Variables

B.2

B.2

$H(s) = \frac{K}{\tau s+1} \rightarrow$ System $\qquad x(t) = u(t) \xrightarrow{\mathcal{L}} X(s) = \frac{1}{s}$

Output: $y(t) = x(t) * h(t) \rightarrow Y(s) = H(s)X(s) = \frac{K}{\tau s+1} \cdot \frac{1}{s}$

① $\frac{K}{s(\tau s+1)} = \frac{A}{s} + \frac{B}{s\tau+1}$

$K = A\tau s + A + Bs$

$A = K \qquad B = -K\tau$

$A\tau + B = 0$

$Y(s) = \frac{K}{s} - \frac{K\tau}{s\tau+1} = \frac{K}{s} - \frac{K}{s+\frac{1}{\tau}} \xrightarrow{\mathcal{L}^{-1}} y(t) = \left(K - Ke^{-\frac{t}{\tau}}\right) u(t)$

$\boxed{y(t) = K\left(1 - e^{-\frac{t}{\tau}}\right) u(t)}$

Figure 13: Step Response of First Order System Calculation to Time Domain

Moving the pole location on the real axis will never yield any oscillations in the unit step output. Complex conjugate poles are needed for a system to output oscillations with a unit step input. To achieve oscillations, the system must be second-order higher with a pair of complex conjugate poles.

C: Closed-Loop vs Open-Loop Control

C.1

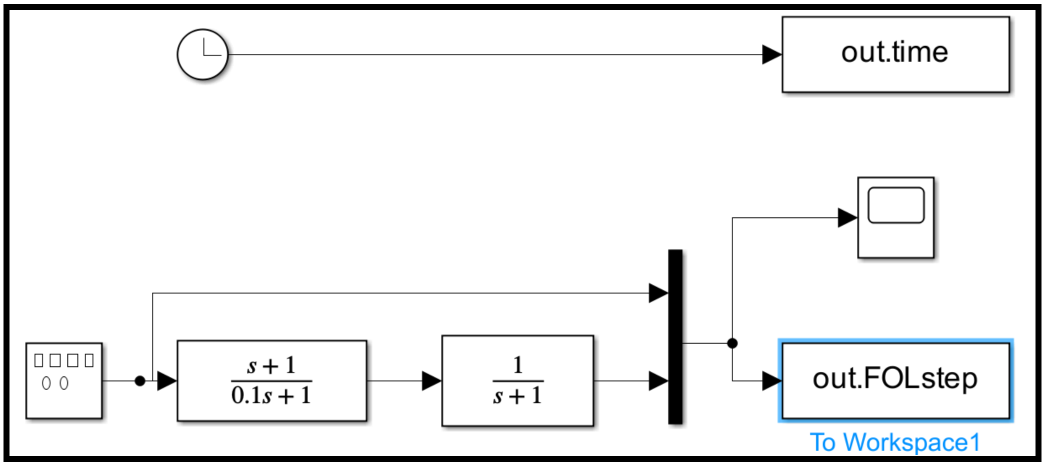


Figure 14: Print of Simulink Model

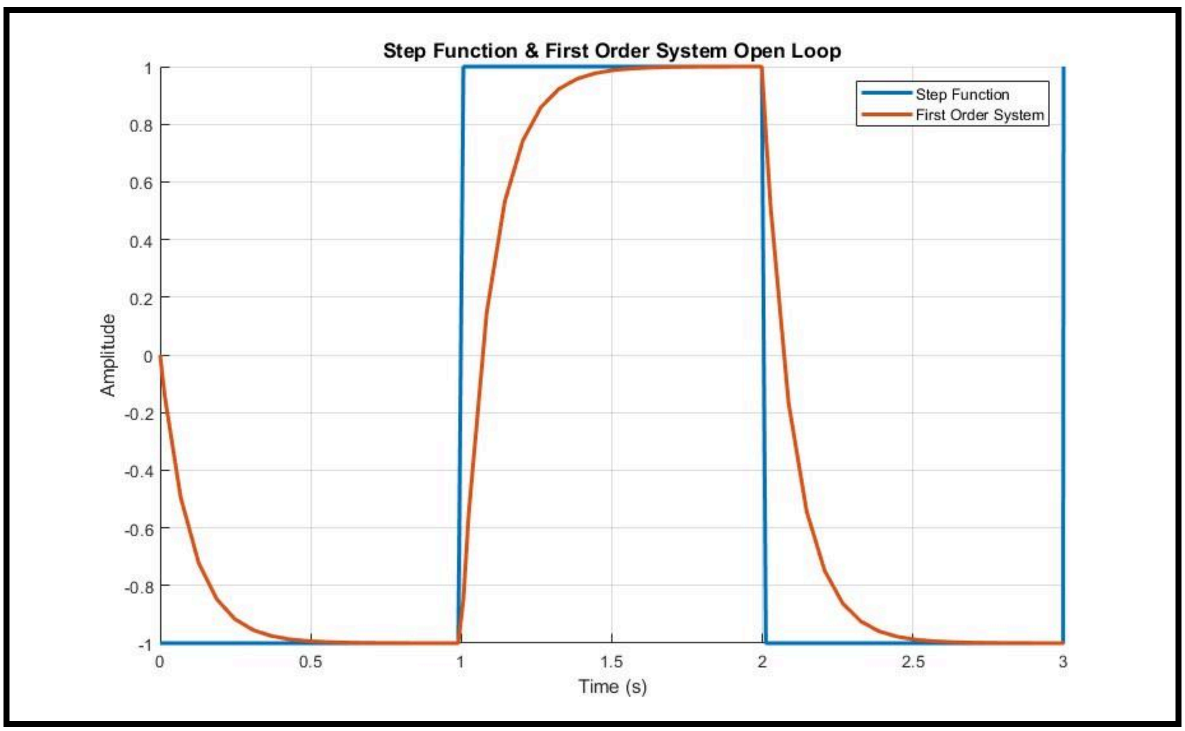


Figure 15: Plotted Reference Input and Output of System

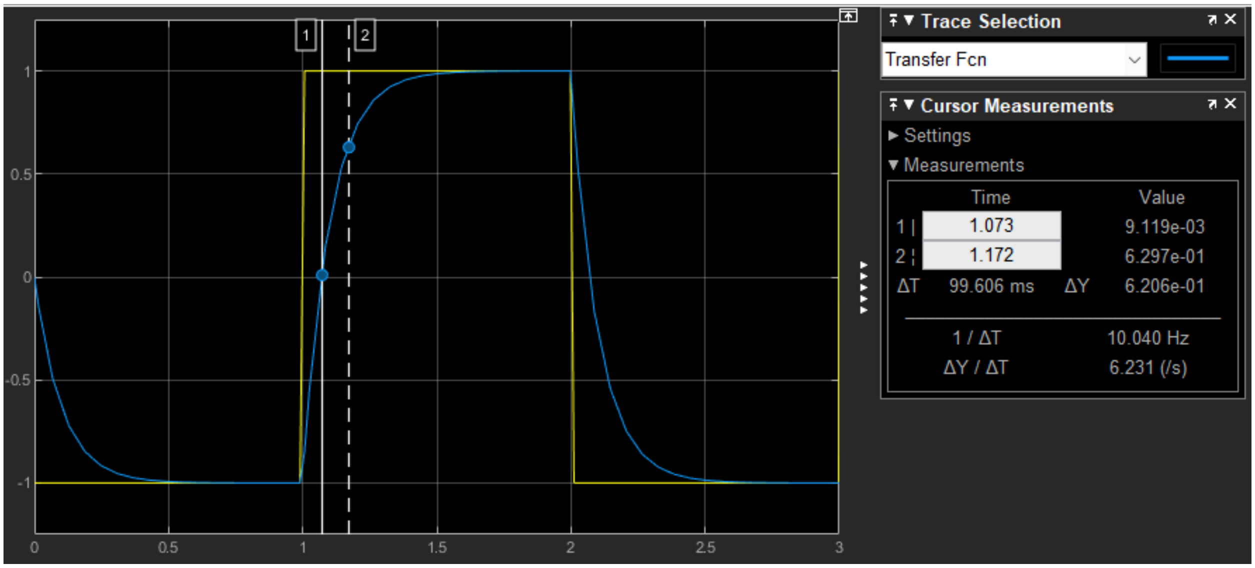


Figure 16: Measured Time Constant $\tau = 1.172 - 1.073 = 0.099$

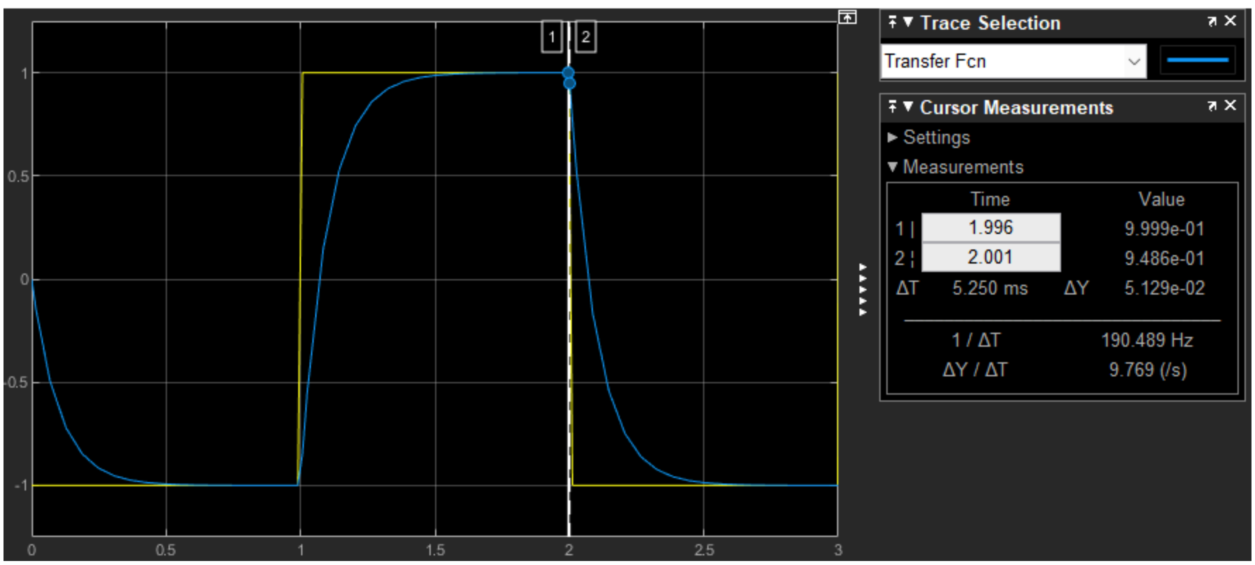


Figure 17: Measured Steady-State $y_{ss} = 0.9999$

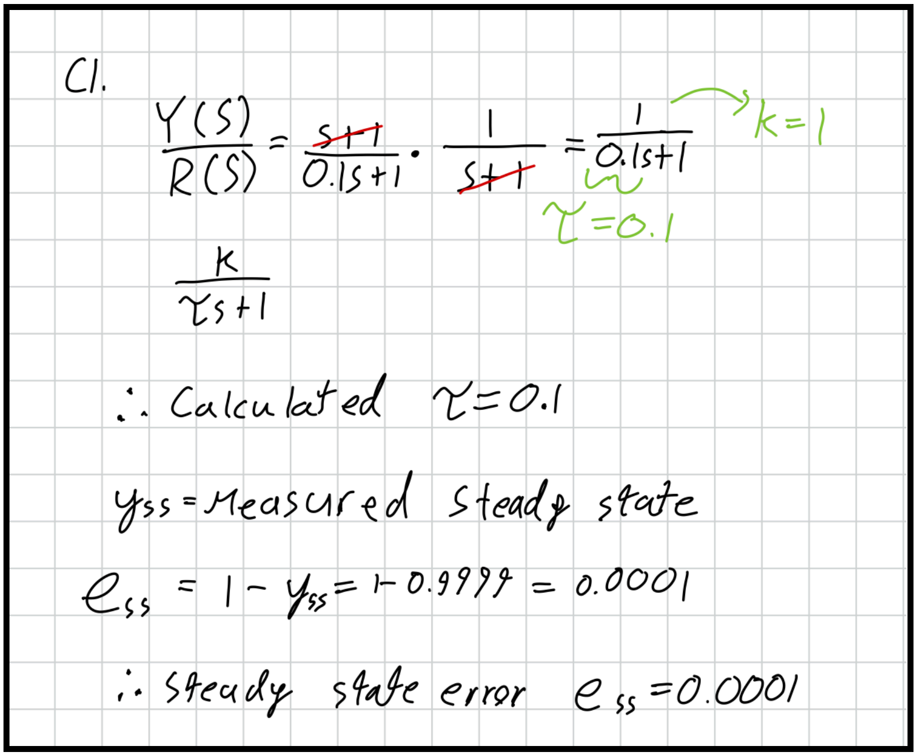


Figure 18: Transfer Function $\frac{Y(s)}{R(s)} = \frac{1}{0.1s+1}$ *Calculated Time Constant* $\tau = 0.1$ *and Steady State Error* $e_{SS} = 0.0001$

C.2

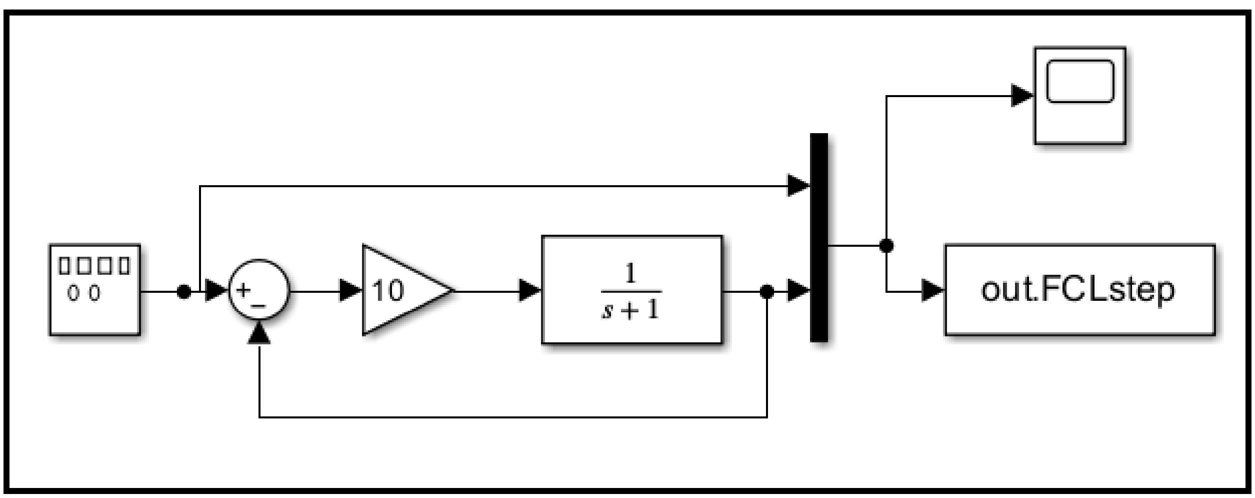


Figure 19: Print of Simulink Model

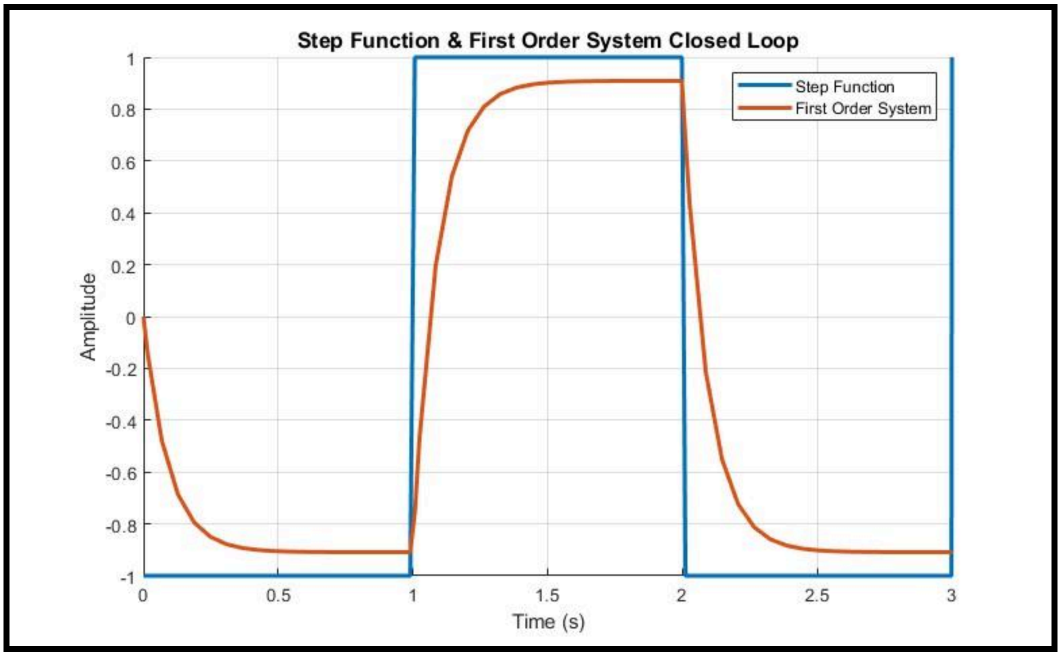


Figure 20: Reference Input and Square Wave Response of System

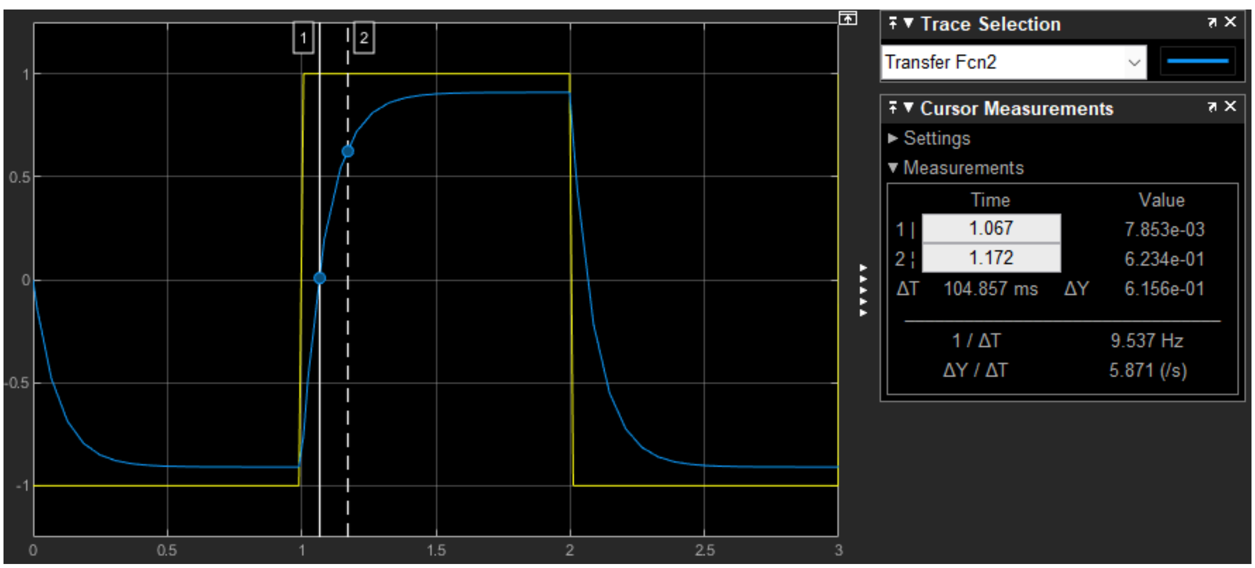


Figure 21: Measured Time Constant $\tau = 1.172 - 1.067 = 0.105$

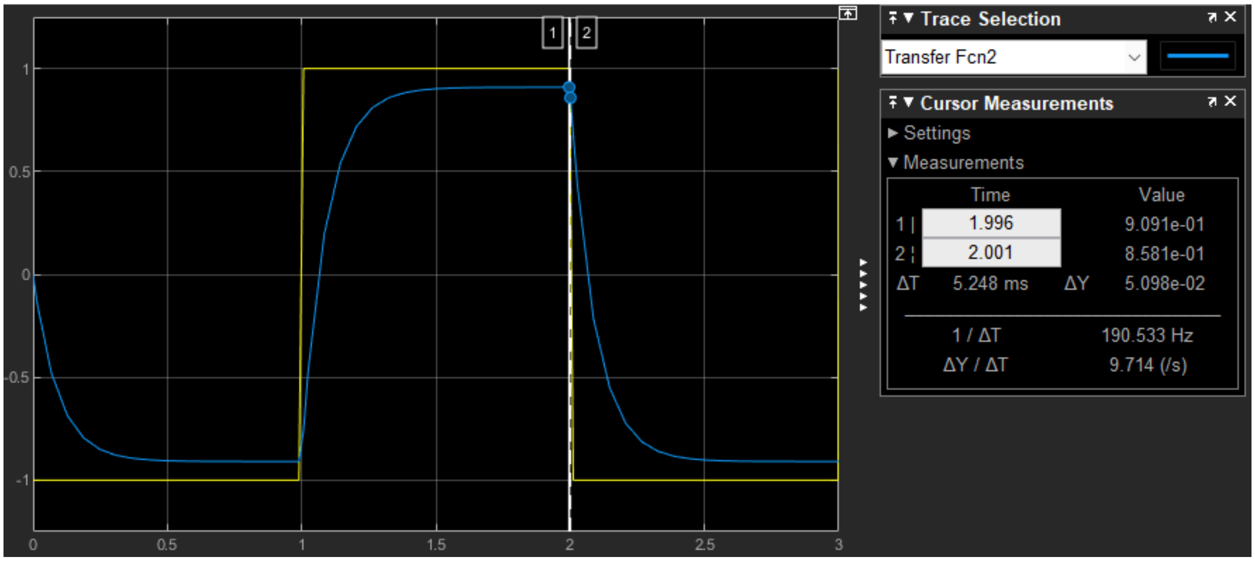


Figure 22: Measured Steady-State $y_{ss} = 0.9091$

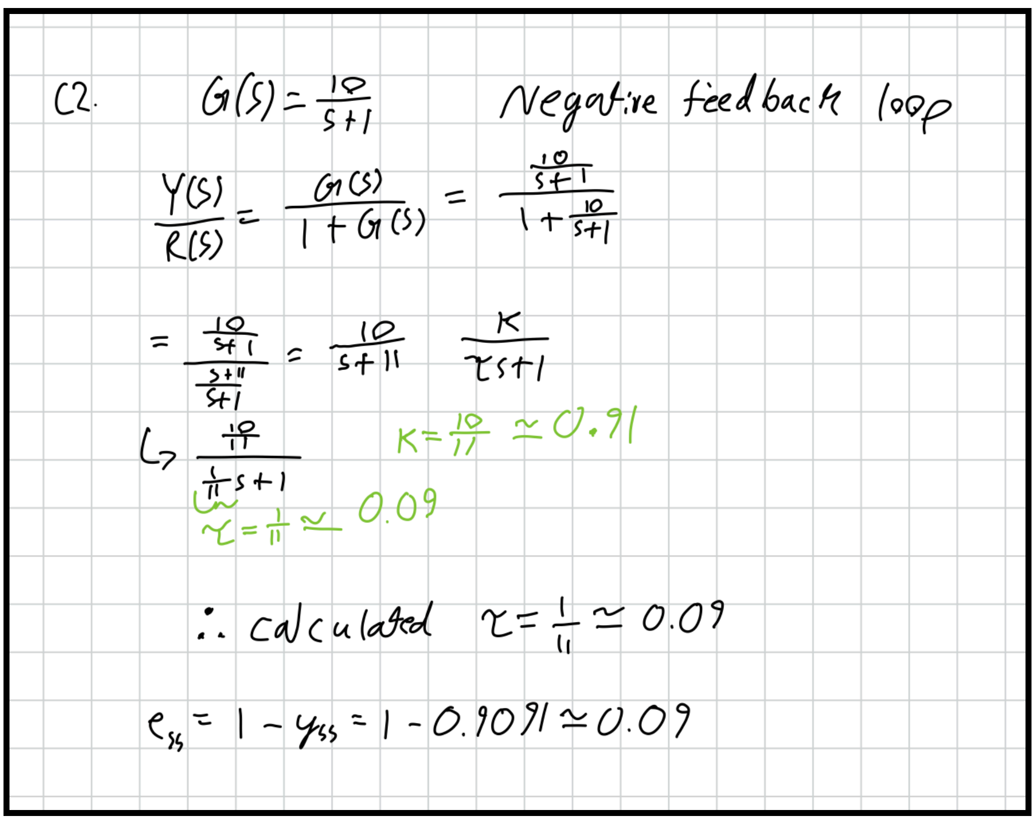


Figure 23: Transfer Function $\frac{Y(s)}{R(s)} = \frac{\frac{10}{11}}{\frac{1}{11}s+1}$ *Calculated Time Constant* $\tau \simeq 0.09$ *and Steady State Error* $e_{ss} = 0.09$

C.3

Table 1: Open-Loop Versus Closed-Loop Approach Comparison Results

	Time constant (τ)	**Steady-state error (e_{SS})**
Open-loop Approach	**0.099s**	**0.0001**
Closed-loop Approach	**0.105s**	**0.09**

The method that should increase the speed exactly 10 times is the open loop approach because the time constant in the theoretical equation of the system is 1/10 of the original system as seen in figure 18. Conversely, the closed-loop has an amplification of 10/11 which introduces a steady-state error. The theoretical final value modifies the time constant value location to 63% of the steady-state which would be $\frac{10}{11} \times 0.63 = 0.57$. When adjusting the cursor of the scope in Figure 21 to 0.57 amplitude, the result is consistent with the calculated theoretical time constant of $\tau \simeq 0.09$. The closed-loop approach takes advantage of the assumption that the time constant is at 63% of the original expected output, which lengthens the time constant closer to the desired $\tau = 0.1$.

C.4

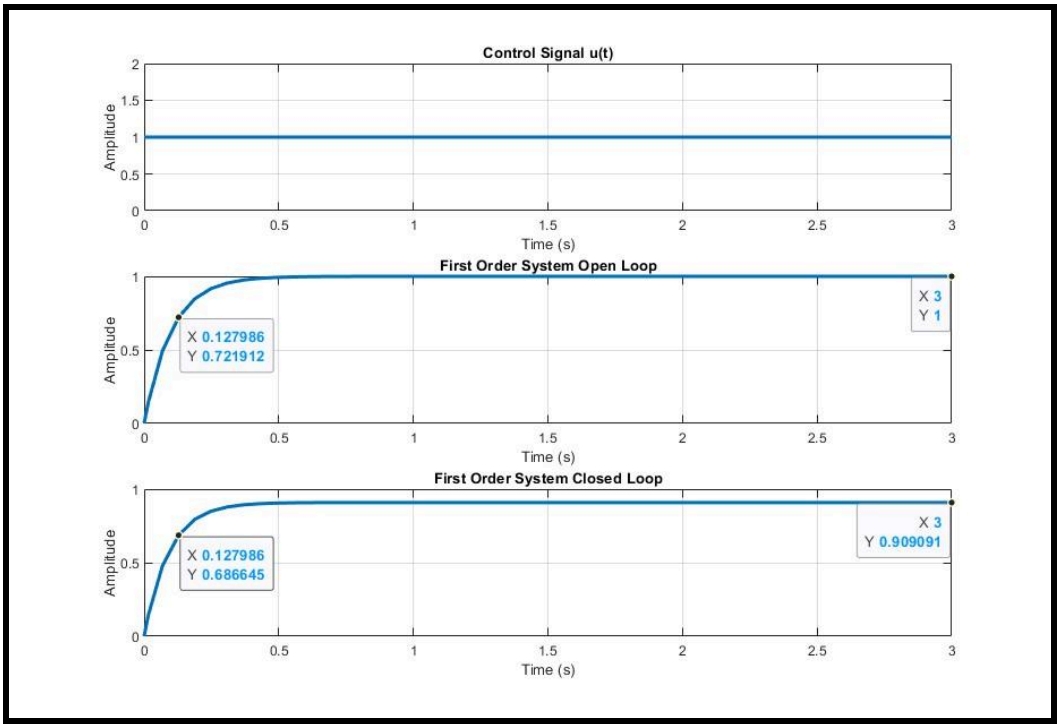


Figure 24: Control Signal Input and Output Responses Plots

The points selected in Figure 24 illustrate the difference between the two systems. The calculated theoretical expected values are similar to the measured values for both systems. The open-loop system with an amplification of $k = 1$ as shown at the steady-state point (3, 1). The steady-state of the closed-loop system has an amplification of $k = 0.909$ as shown at the steady-state point (3, 0.909). Simulink's output to the workspace does not have enough data points to accurately determine the time constant. Although, selecting the same time around the 63% amplitude shows a difference between both systems. This indicates that the response time will vary between the systems, ultimately giving a different output for each system.

C.5

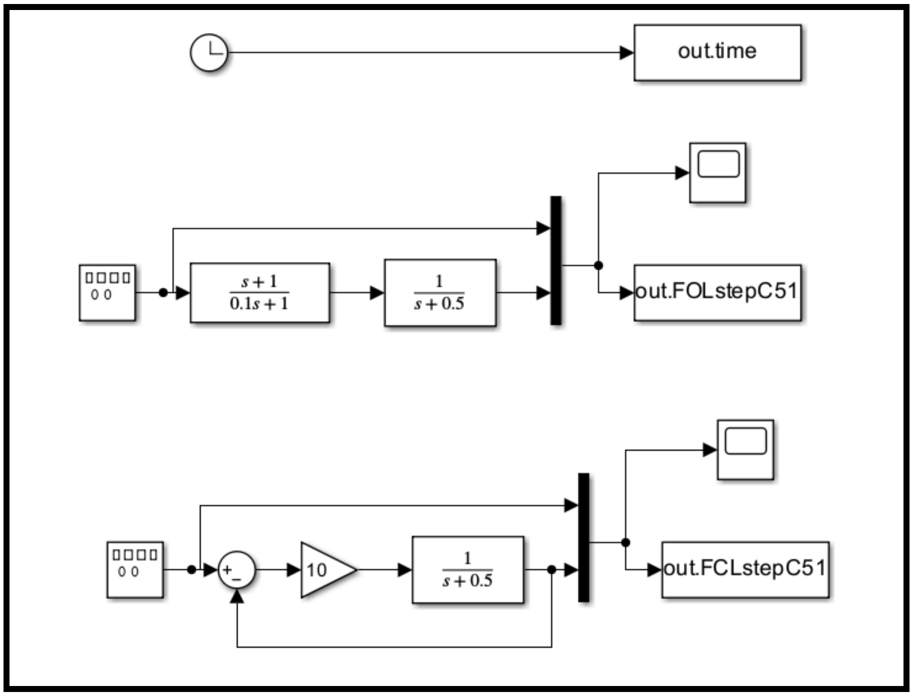


Figure 25: Print of $\frac{1}{s+0.5}$ Simulink Model

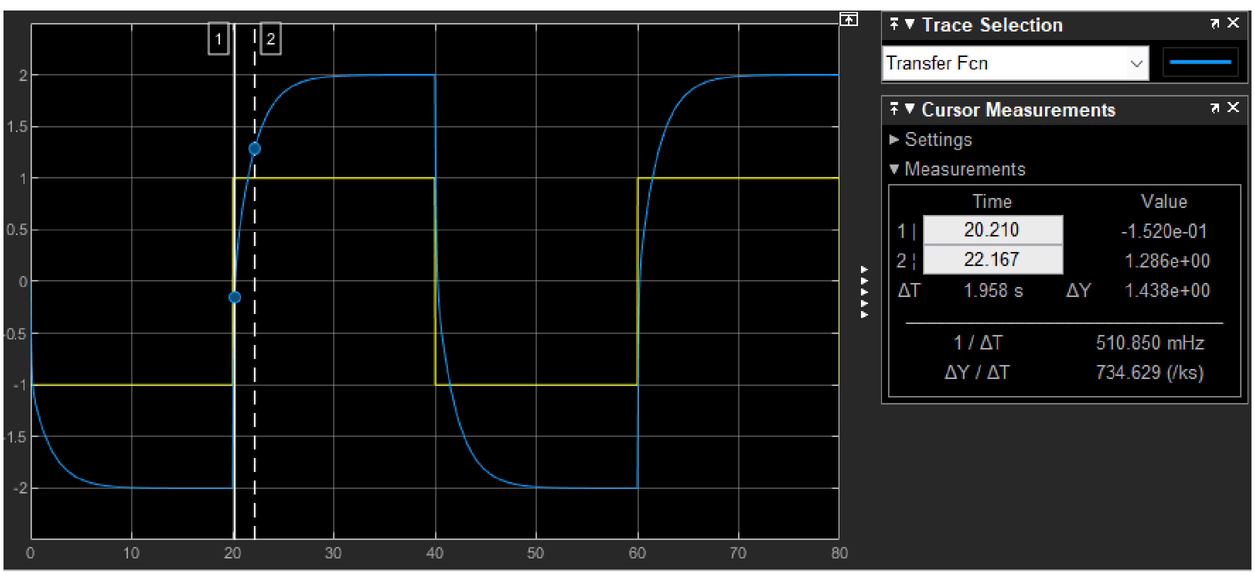


Figure 26: Closed Loop Measured Time Constant at $y_{ss} = 2$, Amplitude

$0.63 \times y_{ss} = 1.26, \ \tau = 1.958$

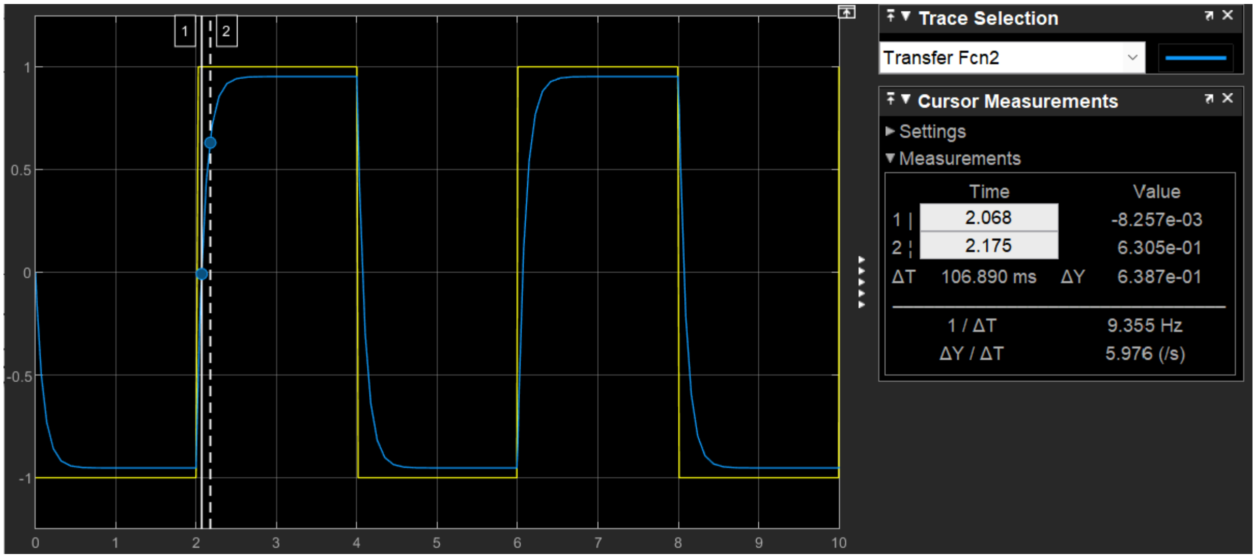


Figure 26: Open Loop Measured Time Constant at $y_{ss} = 0.95$ *Amplitude*

$0.63 \times y_{ss} = 0.6, \ \tau = 0.107$

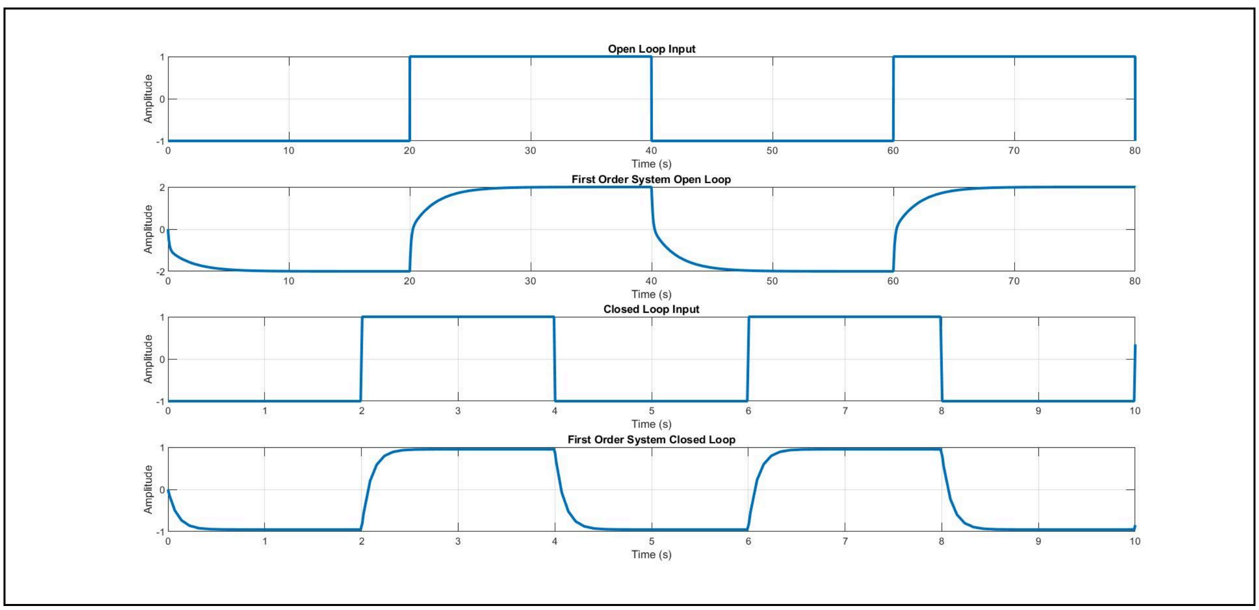


Figure 27: Plotted Reference Inputs and Outputs of Both Systems

Table 2: Measured and Calculated Functions and Values

Main System $\frac{1}{s+0.5}$	$\frac{Y(S)}{R(S)}$	**Time constant (τ)**	**Steady-state error** $(e_{SS} = 1 - y_{SS})$
Open-loop Approach	$\frac{s+1}{0.1s+1} \times \frac{1}{s+0.5}$	**1.985s**	**-1**
Closed-loop Approach	$\frac{\frac{20}{21}}{\frac{2}{21}s+1}$	**0.107s**	**0.05**

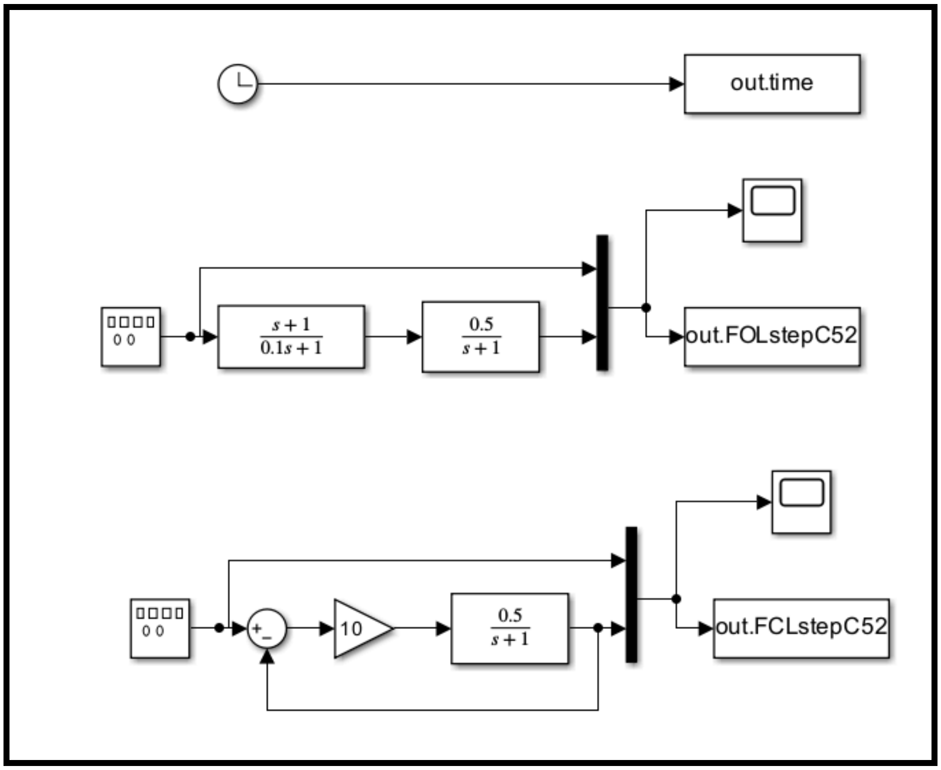


Figure 28: Print of $\frac{0.5}{s+1}$ *Simulink Model*

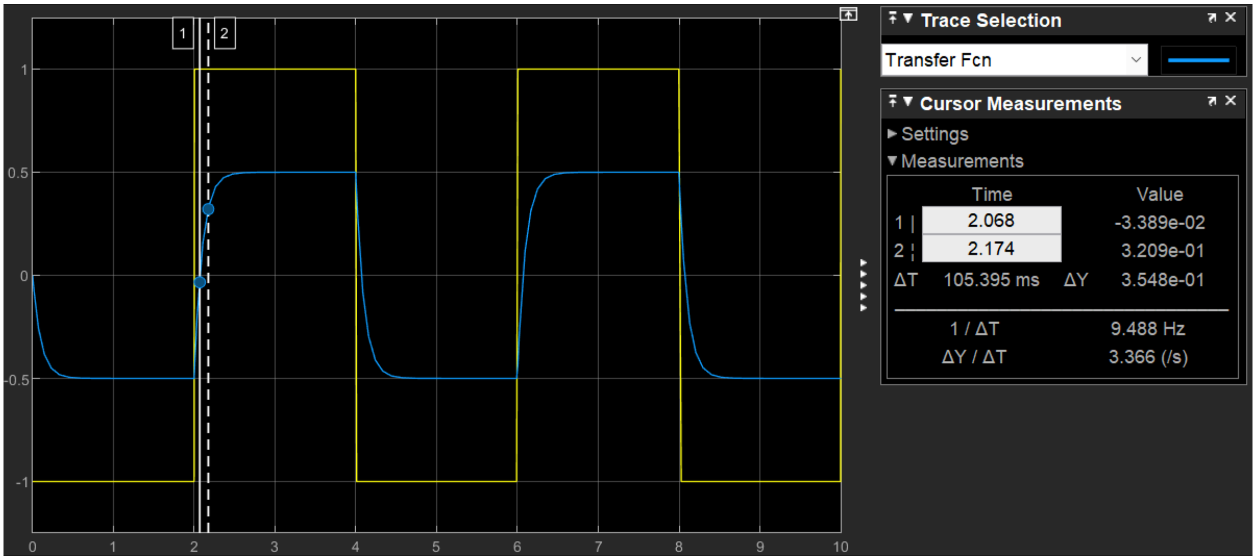


Figure 29: Closed Loop Measured Time Constant at $y_{ss} = 0.5$ *Amplitude*

$$0.63 \times y_{ss} = 0.315, \ \tau = 0.105$$

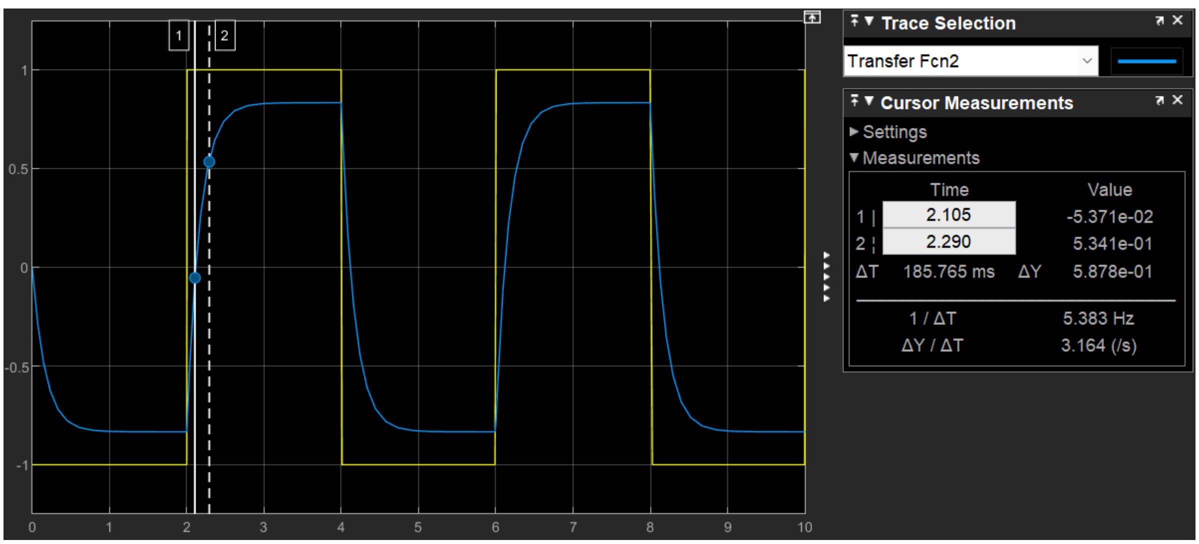


Figure 30: Open Loop Measured Time Constant at $y_{ss} = 0.833$ *Amplitude*

$$0.63 \times y_{ss} = 0.525, \ \tau = 0.186$$

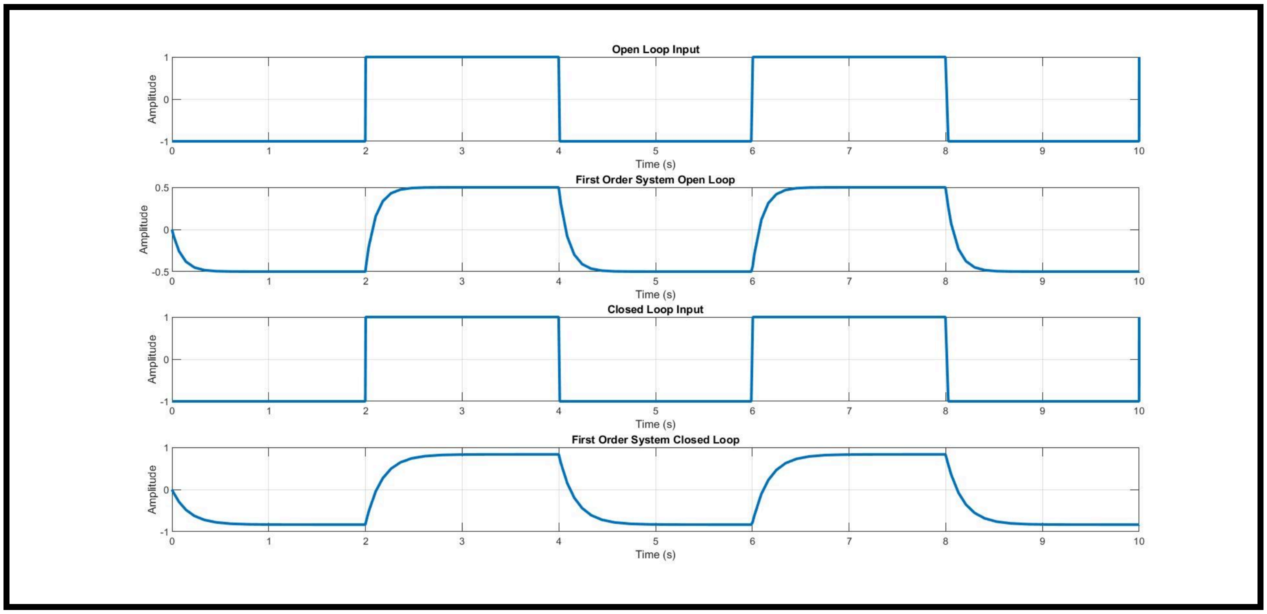


Figure 31: Plotted Reference Inputs and Outputs of Both Systems

Table 3: Measured and Calculated Functions and Values

Main System $\frac{0.5}{s+1}$	$\frac{Y(S)}{R(S)}$	**Time constant (τ)**	**Steady-state error ($e_{SS} = 1 - y_{SS}$)**
Open-loop Approach	$\frac{s+1}{0.1s+1} \times \frac{0.5}{s+1}$	**0.105s**	**0.5**
Closed-loop Approach	$\frac{\frac{5}{s+1}}{\frac{5}{s+1}+1}$	**0.186s**	**0.167**

C.5

From the pole sensitivity test recorded in table 2, the closed-loop approach was more robust to variations. With a closed-loop negative feedback, the transfer function simplifies to $\frac{G(s)}{1+G(s)}$ where G(s) is the modified function with expected variations. When the function approaches a pole, a limit of $\frac{\infty}{1+\infty} \simeq 1$, results in less of an impact on the output.

On the other hand, table 3 shows significant variation in the time constant of the closed-loop approach when gain sensitivity was tested. The closed-loop approach relies on the gain of 10 to adjust the time constant to the same approximate location. In the test the gain is halved to 5, ultimately throwing off the constant by roughly double the desired value.

Depending on the expected variations in the system, a closed loop approach would be more robust or variations in the pole. An open-loop approach would be more robust if the time constant variable needed to be stable but the steady-state error would need to be accounted for because the gain variations would directly affect the output.

PART 2

A: Introduction to High-order Systems

A.1

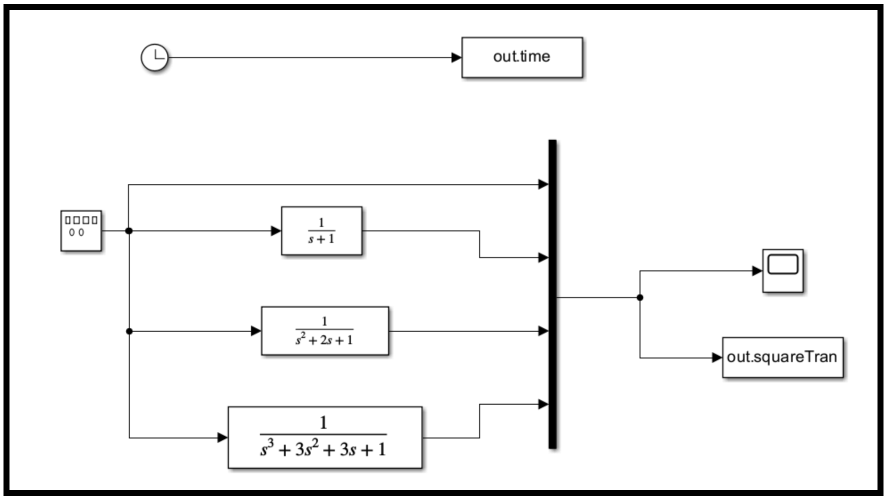


Fig 32: Simulink Model of first second and third order system

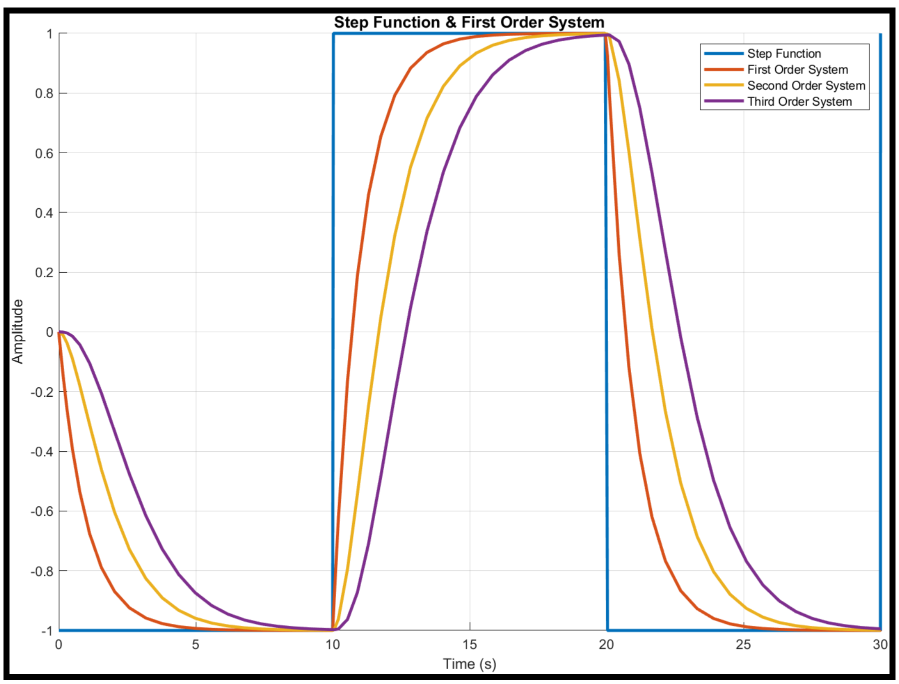


Fig 33: Plot of the square wave input and the responses plotted by the high order open-loop systems.

A.2

The step response of a system depends on the order of the transfer function. One change observed is the time the response takes to settle at its steady state. Higher order systems take a longer time to reach steady state and take longer to respond.

B: Time response of second-order systems

B.1

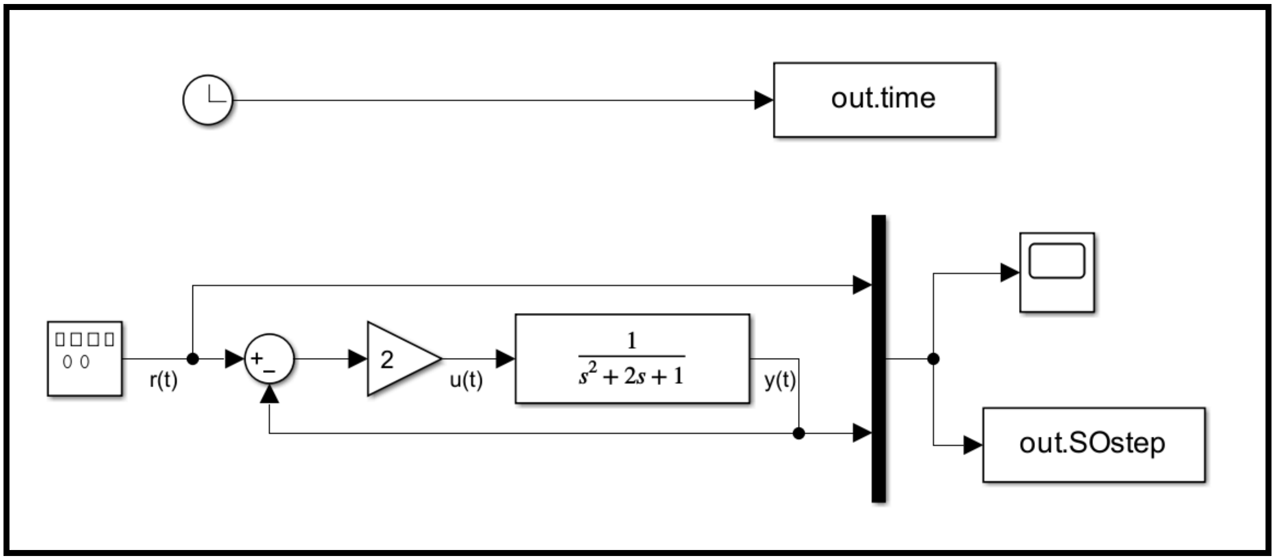


Fig 34: Printed k=2 Simulink Model

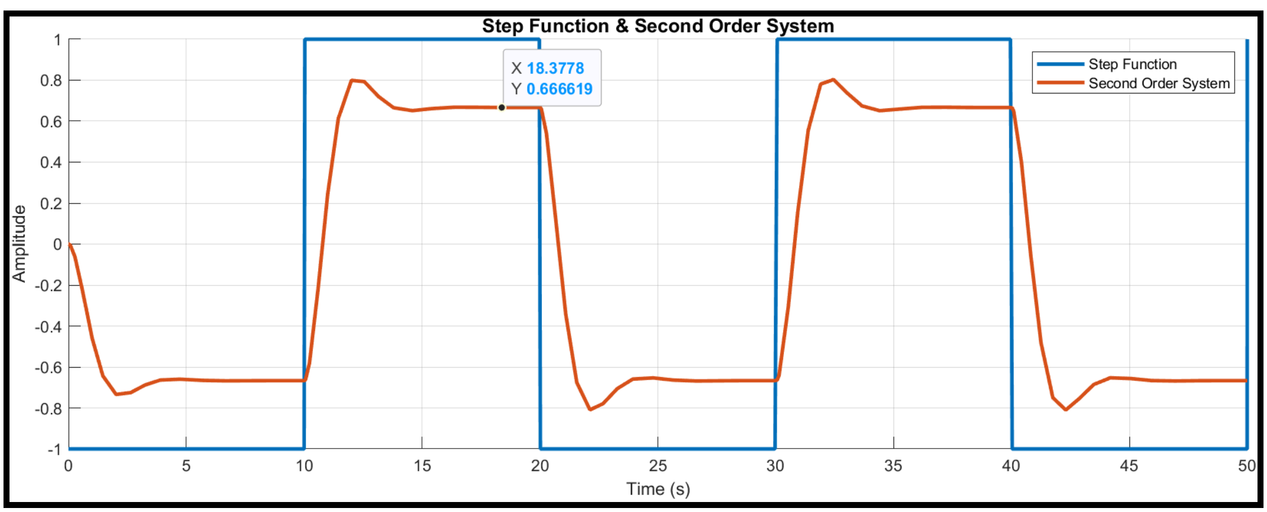


Fig 35: Plotted Reference Input and Output for k = 2

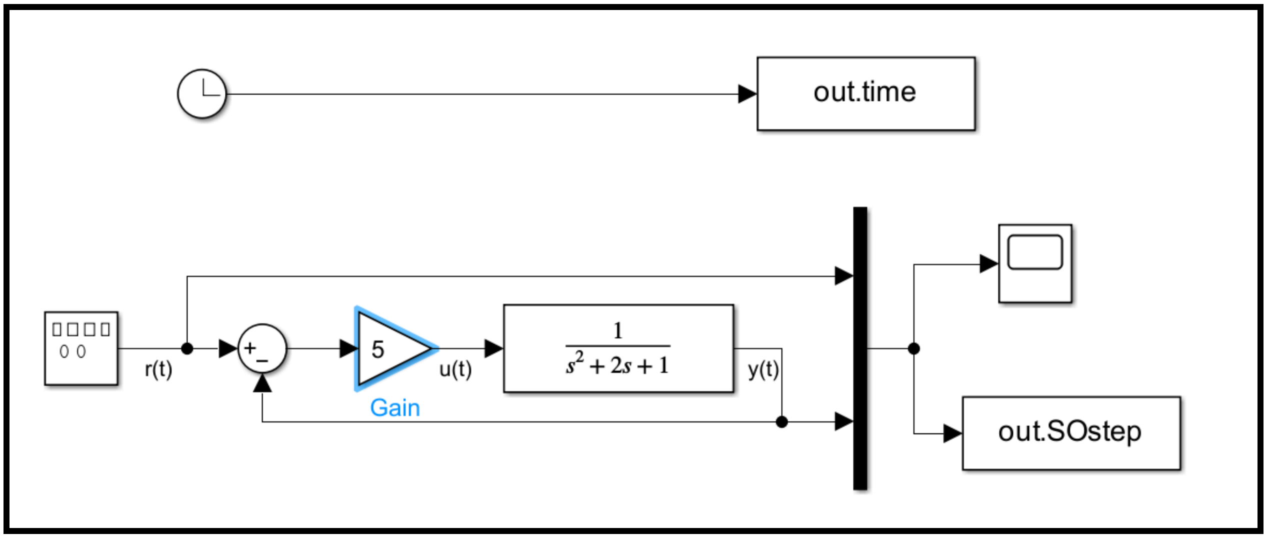


Fig 36: Printed k=5 Simulink Model

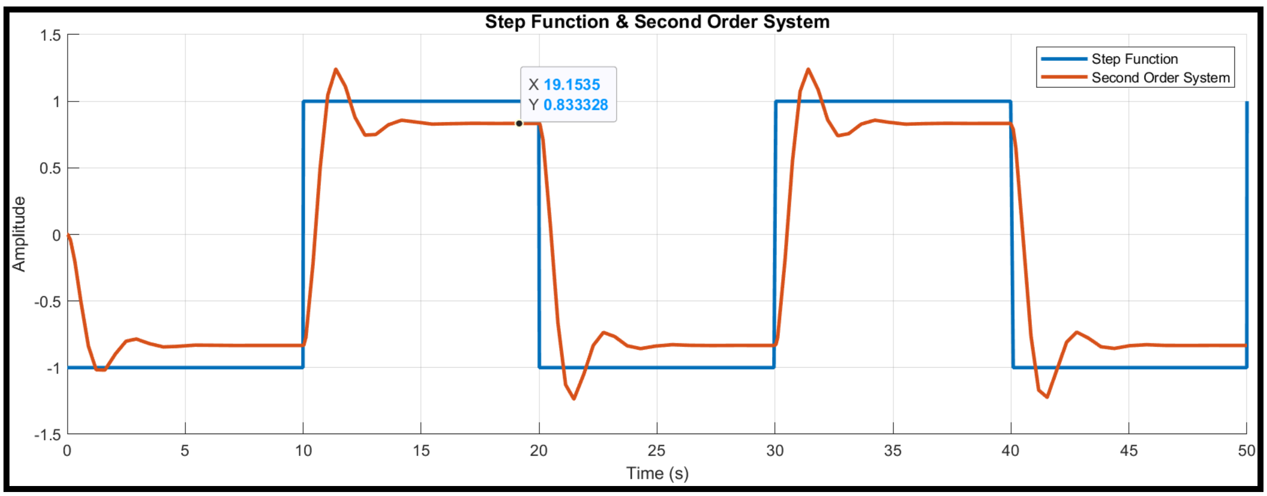


Fig 37: Plotted Reference Input and Output for k = 5

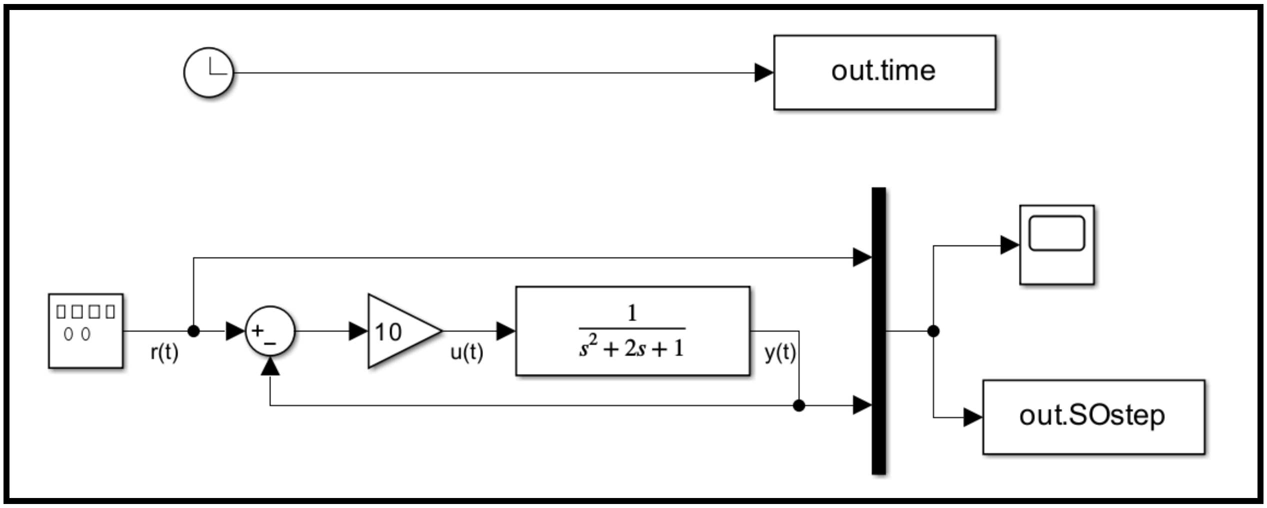


Fig 38: Printed k=10 Simulink Model

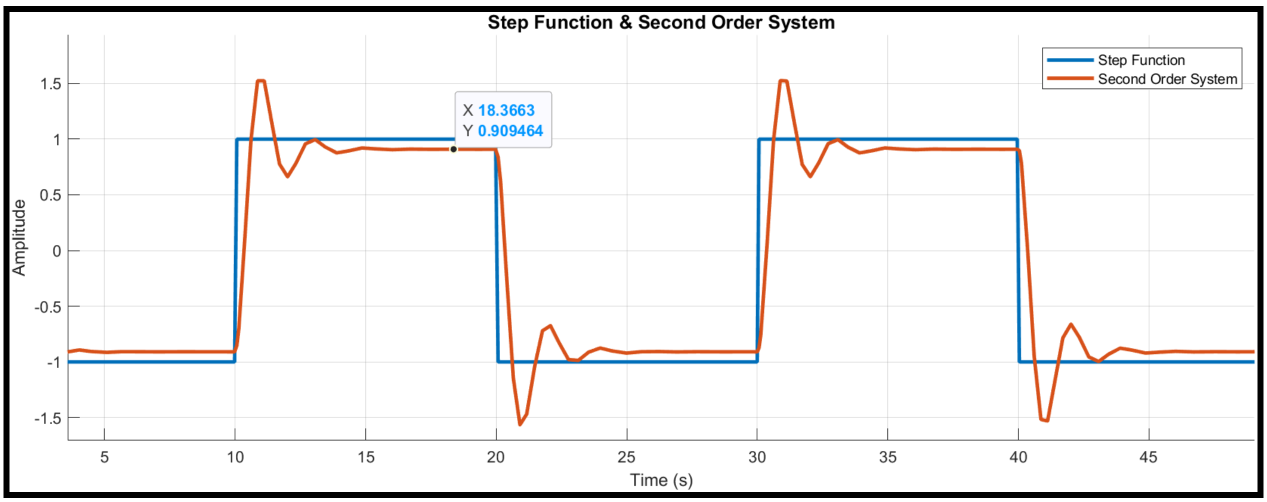
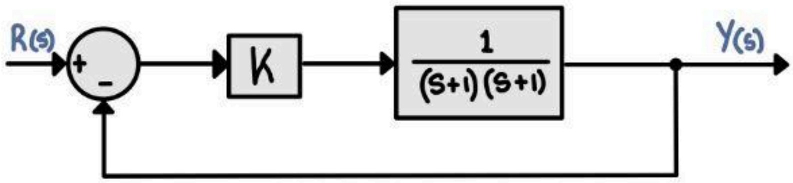
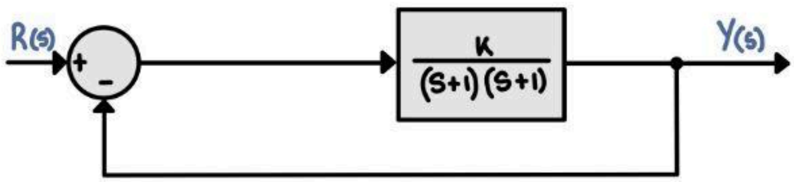


Fig 39: Plotted Reference Input and Output for k = 10

B.2





$$Y(s) = \left[R(s) - Y(s)\right]\left[\frac{1}{(s+1)(s+1)}\right][K]$$

$$Y(s) = \left[R(s) - Y(s)\right]\left[\frac{K}{(s+1)(s+1)}\right]$$

$$Y(s) - Y(s)\frac{K}{(s+1)^2} = R(s)\frac{K}{(s+1)^2}$$

$$\frac{Y(s)}{R(s)}\left(1 + \frac{K}{(s+1)^2}\right) = \frac{K}{(s+1)^2}$$

$$\frac{Y(s)}{R(s)} = \frac{\frac{K}{(s+1)^2}}{1 + \frac{K}{(s+1)^2}} = \frac{\frac{K}{\cancelto{1}{(s+1)^2}}}{\frac{(s+1)^2 + K}{\cancelto{1}{(s+1)^2}}}$$

$$\frac{Y(s)}{R(s)} = \frac{K}{(s+1)^2 + K}$$

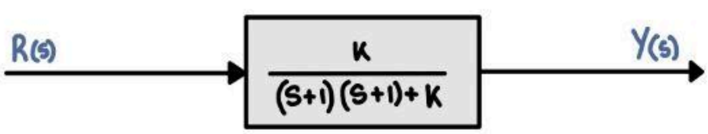


Fig 40: Closed-Loop General Transfer Function

TRANSFER FUNCTIONS

$$K = 2 \rightarrow \frac{Y(s)}{R(s)} = \frac{2}{(s+1)^2 + 2} = \frac{2}{s^2 + 2s + 3}$$

$$K = 5 \rightarrow \frac{Y(s)}{R(s)} = \frac{5}{(s+1)^2 + 5} = \frac{5}{s^2 + 2s + 6}$$

$$K = 10 \rightarrow \frac{Y(s)}{R(s)} = \frac{10}{(s+1)^2 + 10} = \frac{10}{s^2 + 2s + 11}$$

Fig 41: Closed-Loop Transfer Functions at Specified Gains

```
K2 =

  -1.0000 + 1.4142i
  -1.0000 - 1.4142i


K5 =

  -1.0000 + 2.2361i
  -1.0000 - 2.2361i


K10 =

  -1.0000 + 3.1623i
  -1.0000 - 3.1623i
```

Fig 42: Zeros of Transfer Functions

B.3

$K=2$

transfer fcn: $\frac{Y(s)}{R(s)} = \frac{2}{s^2+2s+3}$

general form: $\frac{Y(s)}{R(s)} = \frac{W_n^2}{s^2+2\zeta W_n s+W_n^2}$

Natural Frequency & Damping Factor

$$s^2+2s+3 = s^2+2\zeta W_n s+W_n^2$$

$$W_n^2 = 3 \qquad 2\zeta W_n = 2$$
$$W_n = \sqrt{3} \ldots (1) \qquad \zeta W_n = 1$$
$$\zeta = \frac{1}{\sqrt{3}} \ldots (2)$$

$$t_{max} = \frac{\pi}{W_n\sqrt{1-\zeta^2}} = 2.22\,s$$

$$y_{max}[2.22] = 0.739$$

Rise Time

as $0<\zeta<1$;

$$t_r = \frac{0.8+2.5\zeta}{W_n} = \frac{0.8+2.5(1/\sqrt{3})}{\sqrt{3}} = 1.295\,s$$

Settling Time

as $0<\zeta<0.69$

$$t_s \cong \frac{3.2}{\zeta W_n} = \frac{3.2}{1} = 3.2 \text{ seconds}$$

Max Overshoot

Percentage maximum overshoot

$$\%O.S. = 100\,e^{-\pi\zeta/\sqrt{1-\zeta^2}}$$

$$\%O.S. = 100\,e^{-\pi\left(\frac{1}{\sqrt{3}}\right)/\sqrt{1-\left(\frac{1}{\sqrt{3}}\right)^2}}$$

$$= 100\,e^{-1.814/0.816}$$

$$= 10.82\,\%$$

Steady State Error

$$Y_{ss} = 0.667$$

$$|1-Y_{ss}| = 0.333 = 33.3\%\ e_{ss}$$

Fig 43: Natural Frequency, Damping Factor, Rise Time, Percentage of Maximum Overshoot, Settling Time and Steady-State Error Calculations for k = 2

K=5

transfer fcn: $\dfrac{Y(s)}{R(s)} = \dfrac{5}{s^2+2s+6}$

general form: $\dfrac{Y(s)}{R(s)} = \dfrac{W_n^2}{s^2+2\zeta W_n s+W_n^2}$

NATURAL FREQUENCY & DAMPING FACTOR

$$s^2+2s+6 = s^2+2\zeta W_n s+W_n^2$$

$W_n^2 = 6$

$W_n = \sqrt{6}$ ...①

$2\zeta W_n = 2$

$\zeta W_n = 1$

$\zeta = \dfrac{1}{\sqrt{6}}$ ...②

RISE TIME

as $0<\zeta<1$;

$$t_r = \frac{0.8+2.5\zeta}{W_n} = \frac{0.8+2.5(1/\sqrt{6})}{\sqrt{6}} = 0.743 \text{ s}$$

SETTLING TIME

as $0<\zeta<0.69$

$$t_s \cong \frac{3.2}{\zeta W_n} = \frac{3.2}{1} = 3.2 \text{ seconds}$$

MAX OVERSHOOT

Percentage maximum overshoot

$$\%\,O.S. = 100\, e^{-\pi\zeta/\sqrt{1-\zeta^2}}$$

$$\%\,O.S. = 100\, e^{-\pi\left(\frac{1}{\sqrt{6}}\right)/\sqrt{1-\left(\frac{1}{\sqrt{6}}\right)^2}}$$

$$= 100\, e^{-1.283/0.913}$$

$$= 24.53\,\%$$

STEADY STATE ERROR

$Y_{ss} = 0.833$

$|1-Y_{ss}| = 0.167 = 16.7\,\%\ e_{ss}$

Fig 44: Natural Frequency, Damping Factor, Rise Time, Percentage of Maximum Overshoot, Settling Time and Steady-State Error Calculations for k = 5

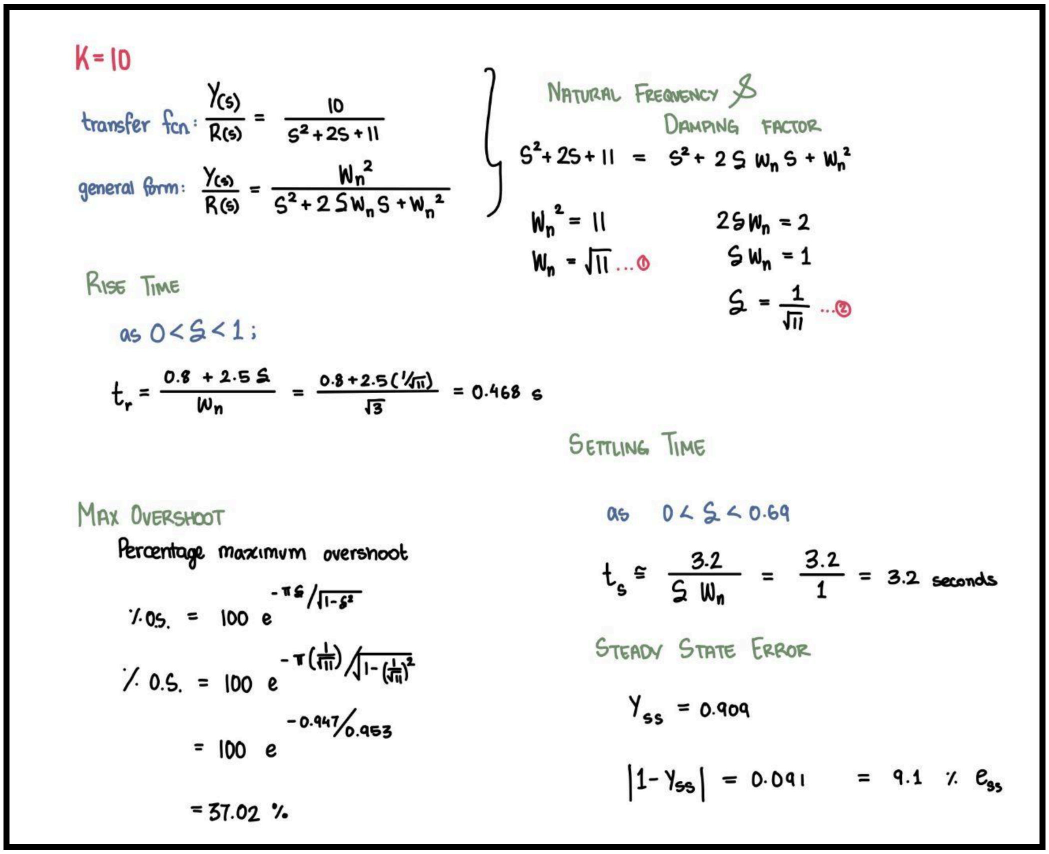


Fig 45: Natural Frequency, Damping Factor, Rise Time, Percentage of Maximum Overshoot, Settling Time and Steady-State Error Calculations for k = 10

Table 4: Tabulated Calculated Variables

	Natural Frequency ω_n	**Damping Factor** ζ	**Rise time** t_r	**Max Overshoot** $\%O.S.$	**Settling Time** t_s	**Steady-state Error** e_{ss}
K = 2	$\sqrt{3}$	$\frac{1}{\sqrt{3}}$	1.295	10.82 %	3.2	33.3 %
K = 5	$\sqrt{6}$	$\frac{1}{\sqrt{6}}$	0.743	24.53 %	3.2	16.7 %
K = 10	$\sqrt{11}$	$\frac{1}{\sqrt{11}}$	0.468	37.02 %	3.2	9.1 %

B.4

The proportional gain affects all the calculated time domain specifications besides settling time because the oscillating nature of a second-order system would still require the same amount of oscillations regardless of their amplitude. As shown in the calculations, the settling time is related to the multiplication of the natural frequency and the dampening factor which will always be 1 in this system configuration.

The rise time decreases as the gain increases because the amplitude from 10% to 90% of the unit input is reached much quicker from the lower dampening factor and higher natural frequency. The calculation is related to dividing the dampening factor by the natural frequency.

The max overshoot percentage increases because the dampening factor has a natural exponential relationship to the peak value as seen in the equation $100e^{\frac{-\pi\varsigma}{\sqrt{1-\varsigma^2}}}$.

The steady-state error can be modelled by the equation $e_{SS} = \lim_{s \to 0} \frac{1}{1+G(s)}$ in this unity feedback system which when the gain increases, the value of $\lim_{s \to 0} G(s)$ gets larger, decreasing the overall error.

B.5

There is no limitation for increasing the gain in terms of stability. The only way for a system to become unstable is to shift its poles to the positive side of the imaginary axis. The final general calculation in figure 40 shows that this would never happen because in the denominator, $(s + 1)(s + 1) + k$, the positive k would never make the poles shift past the imaginary axis.

C: Time response of Third-order systems

C.1

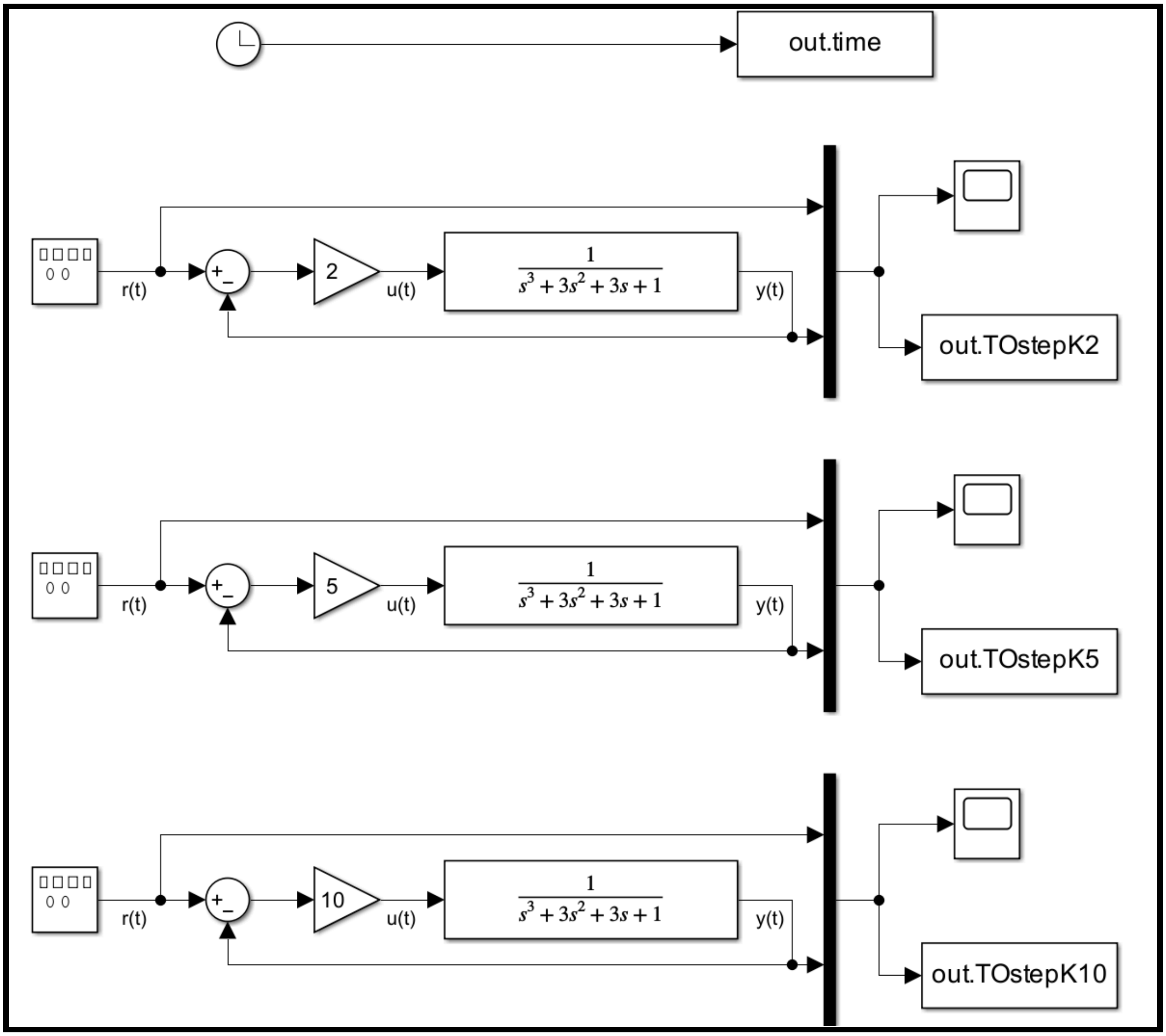


Fig 46: Printed k=2, 5, 10 Simulink Model for third order systems

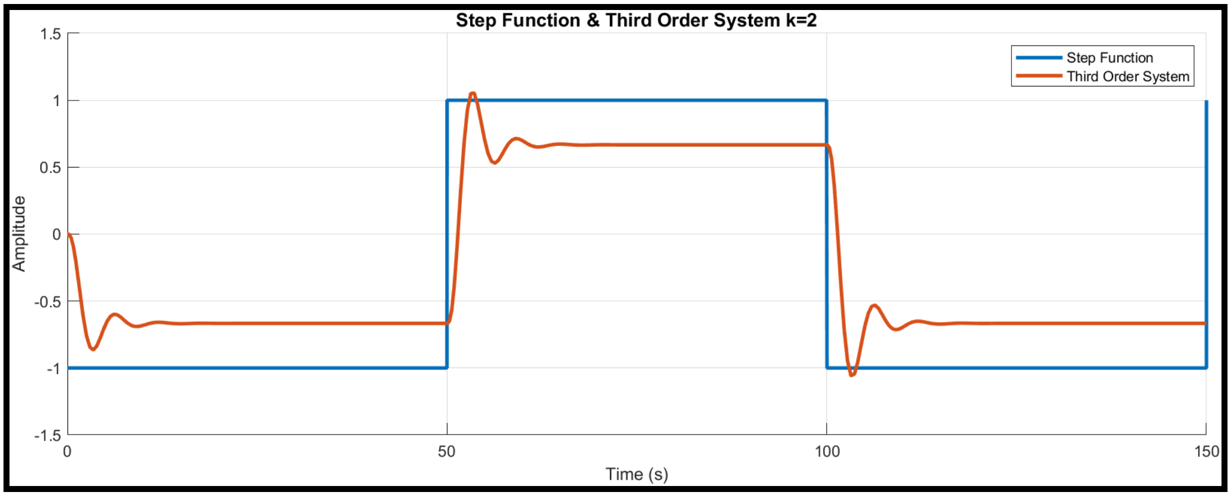


Fig 47: Plotted Input and Output for k = 2 for a Third order system

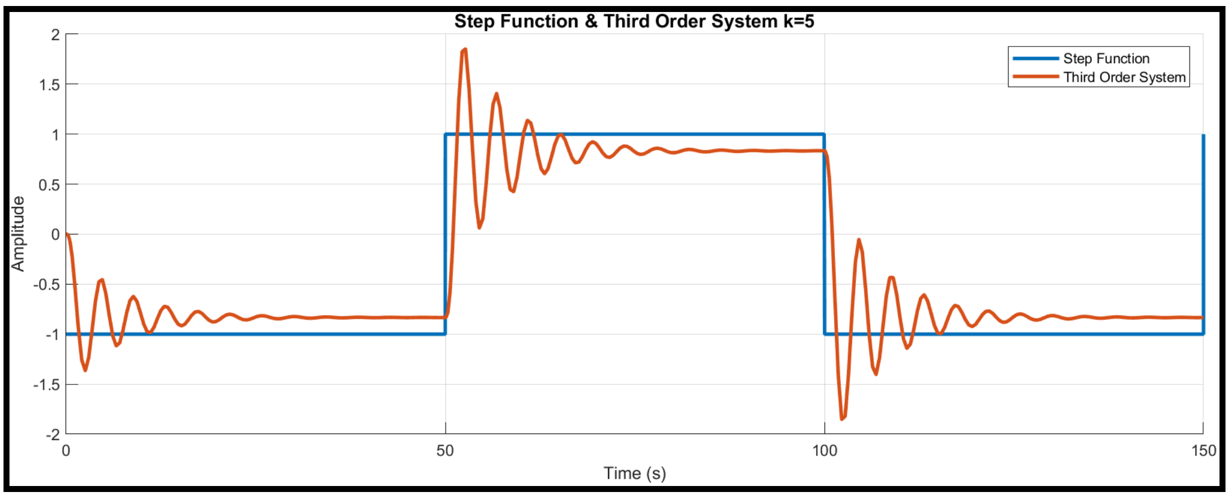


Fig 48: Plotted Input and Output for k = 5 for a Third order system

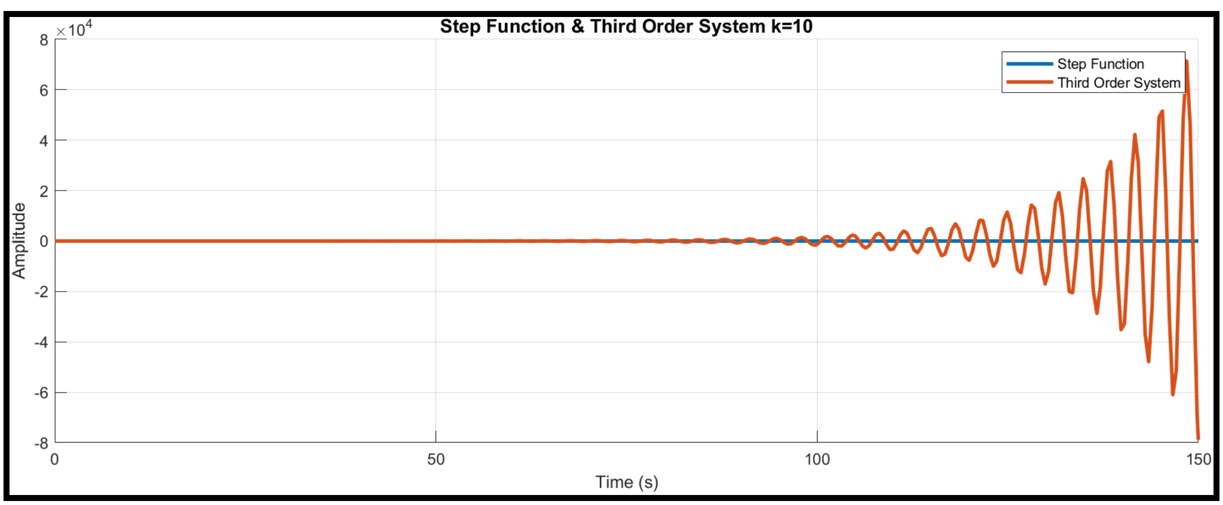
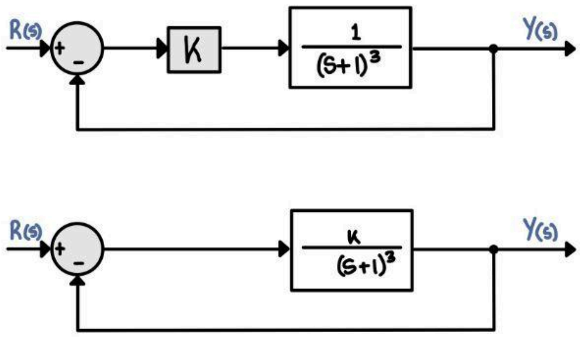


Fig 49: Plotted Input and Output for k = 10 for a Third order system

C.2



$$Y(s) = [R(s) - Y(s)]\left[\frac{1}{(s+1)^3}\right][K]$$

$$Y(s) = [R(s) - Y(s)]\left[\frac{K}{(s+1)^3}\right]$$

$$Y(s) - Y(s)\frac{K}{(s+1)^3} = R(s)\frac{K}{(s+1)^3}$$

$$\frac{Y(s)}{R(s)}\left(1 + \frac{K}{(s+1)^3}\right) = \frac{K}{(s+1)^3}$$

$$\frac{Y(s)}{R(s)} = \frac{\frac{K}{(s+1)^3}}{1 + \frac{K}{(s+1)^3}} = \frac{\frac{K}{\cancel{(s+1)^3}}}{\frac{(s+1)^3 + K}{\cancel{(s+1)^3}}}$$

$$\frac{Y(s)}{R(s)} = \frac{K}{(s+1)^3 + K}$$

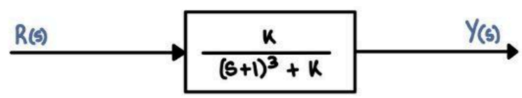


Fig 49: Closed-Loop General Transfer Function for third order system

TRANSFER FUNCTIONS

$$K = 2 \rightarrow \frac{Y(s)}{R(s)} = \frac{2}{(s+1)^3 + 2} = \frac{2}{s^3+3s^2+3s+3}$$

$$K = 5 \rightarrow \frac{Y(s)}{R(s)} = \frac{5}{(s+1)^3 + 5} = \frac{5}{s^3+3s^2+3s+6}$$

$$K = 10 \rightarrow \frac{Y(s)}{R(s)} = \frac{10}{(s+1)^3 + 10} = \frac{10}{s^3+3s^2+3s+11}$$

Fig 50: Closed-Loop Transfer Functions at Specified Gains

```
K2 =

  -2.2599 + 0.0000i
  -0.3700 + 1.0911i
  -0.3700 - 1.0911i


K5 =

  -2.7100 + 0.0000i
  -0.1450 + 1.4809i
  -0.1450 - 1.4809i


K10 =

  -3.1544 + 0.0000i
   0.0772 + 1.8658i
   0.0772 - 1.8658i
```

Fig 51: Zeros of Transfer Functions

C.3

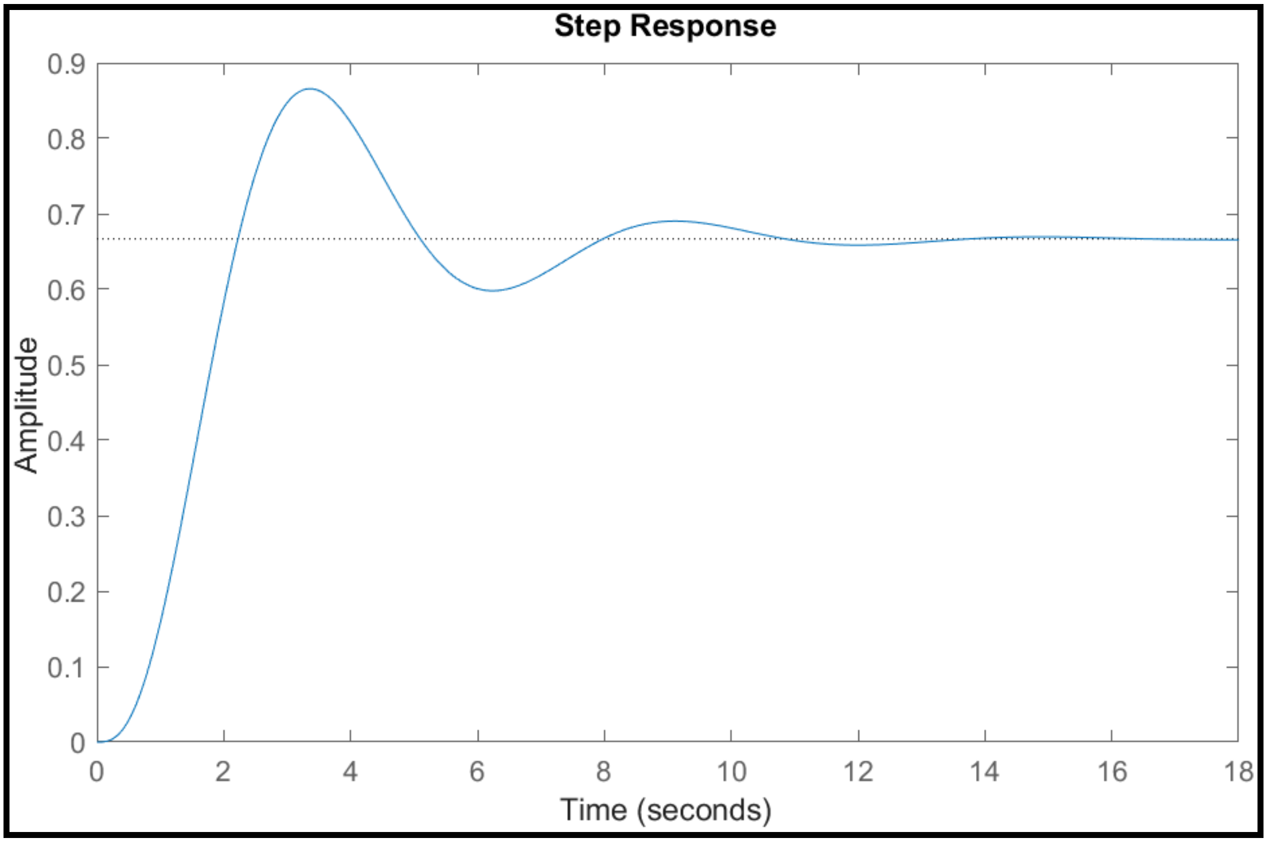
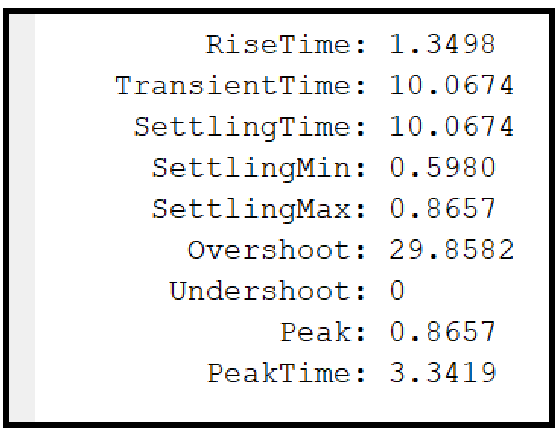


Fig 52: Step response of third order system (k=2) plotted using stepplot



```
      RiseTime: 1.3498
 TransientTime: 10.0674
  SettlingTime: 10.0674
   SettlingMin: 0.5980
   SettlingMax: 0.8657
     Overshoot: 29.8582
    Undershoot: 0
          Peak: 0.8657
      PeakTime: 3.3419
```

Fig 53: Step response info of third order system (k=2) plotted using stepinfo

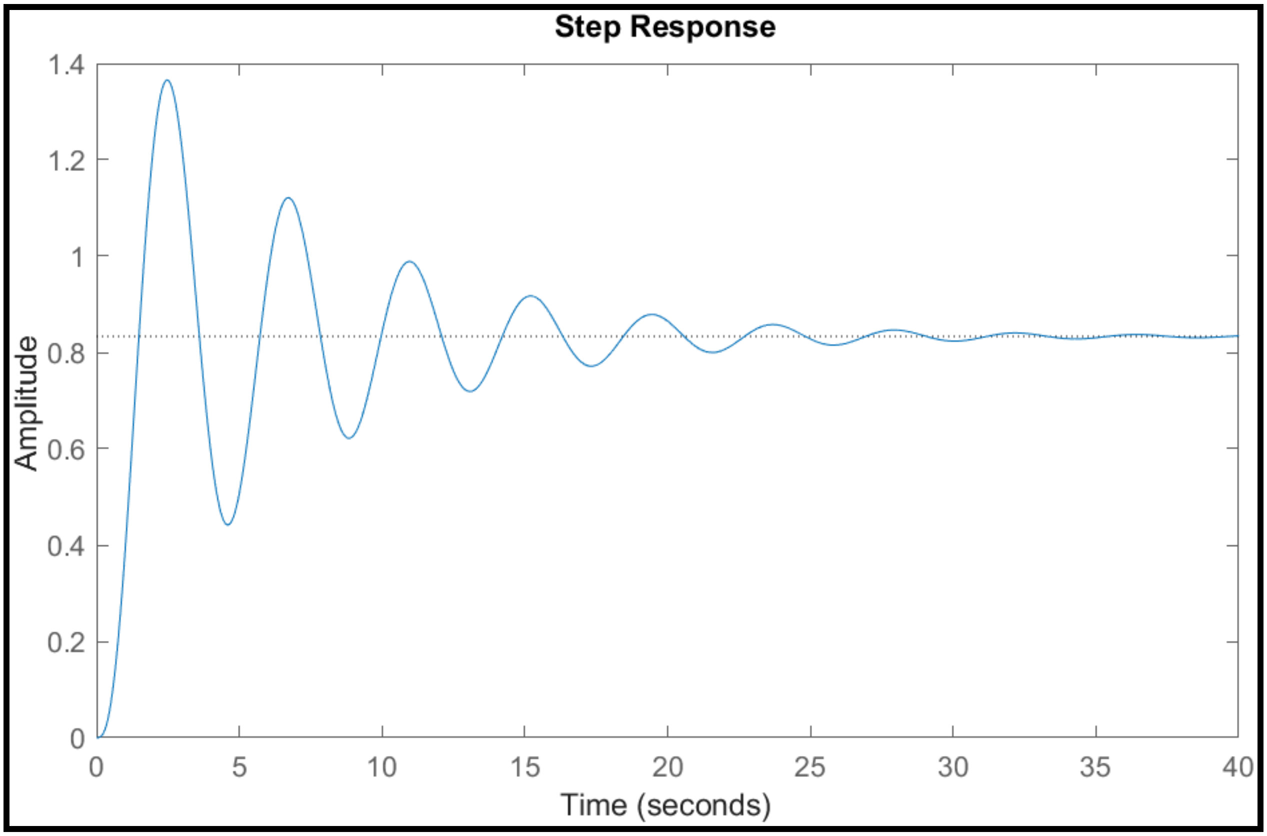


Fig 54: Step response of third order system (k=5) plotted using stepplot

```
        RiseTime: 0.8656
   TransientTime: 26.0796
    SettlingTime: 26.0796
     SettlingMin: 0.4418
     SettlingMax: 1.3656
       Overshoot: 63.8735
      Undershoot: 0
            Peak: 1.3656
        PeakTime: 2.4810
```

Fig 55: Step response info of third order system (k=5) plotted using stepinfo

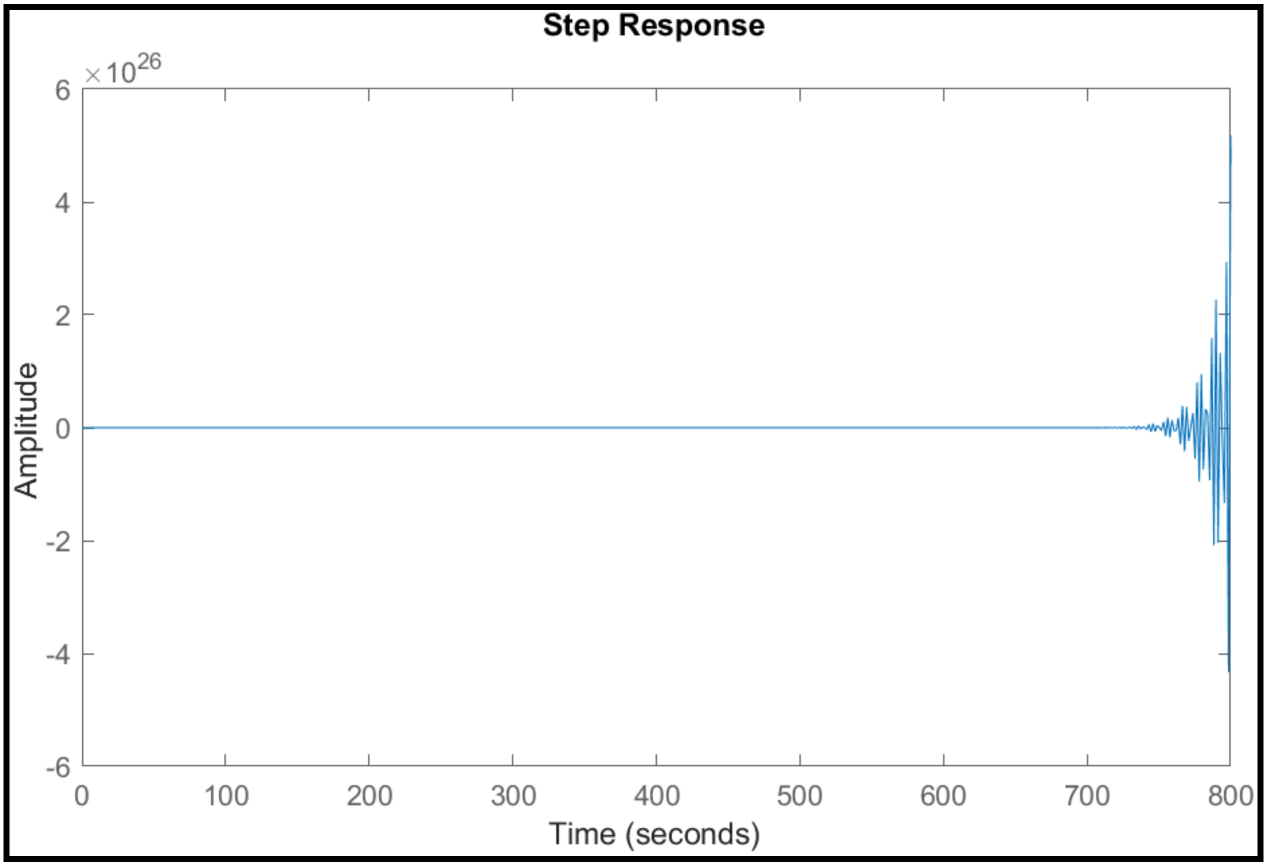


Fig 56: Step response of third order system (k=10) plotted using stepplot

```
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```

Fig 57: Step response info of third order system (k=10) plotted using stepinfo

Table 5: Tabulated Calculated Variables

	Rise time t_r	**Max Overshoot** $\%O.S.$	**Settling Time** t_s	**Steady-state Error** e_{ss}
K = 2	1.35	29.86 %	10.07	33.3 %
K = 5	0.87	63.87 %	26.08	16.7 %
K = 10	NaN	NaN	NaN	inf %

C.4

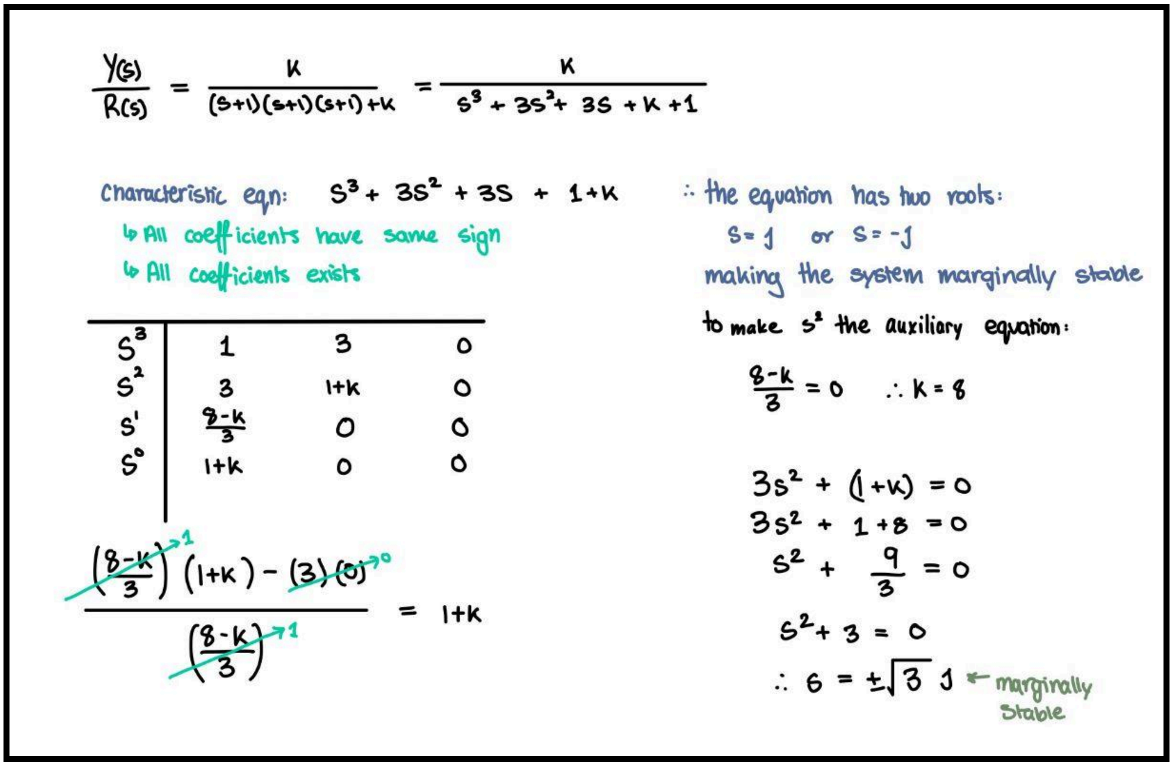


Fig 58: Calculation for the Marginal Stability Gain

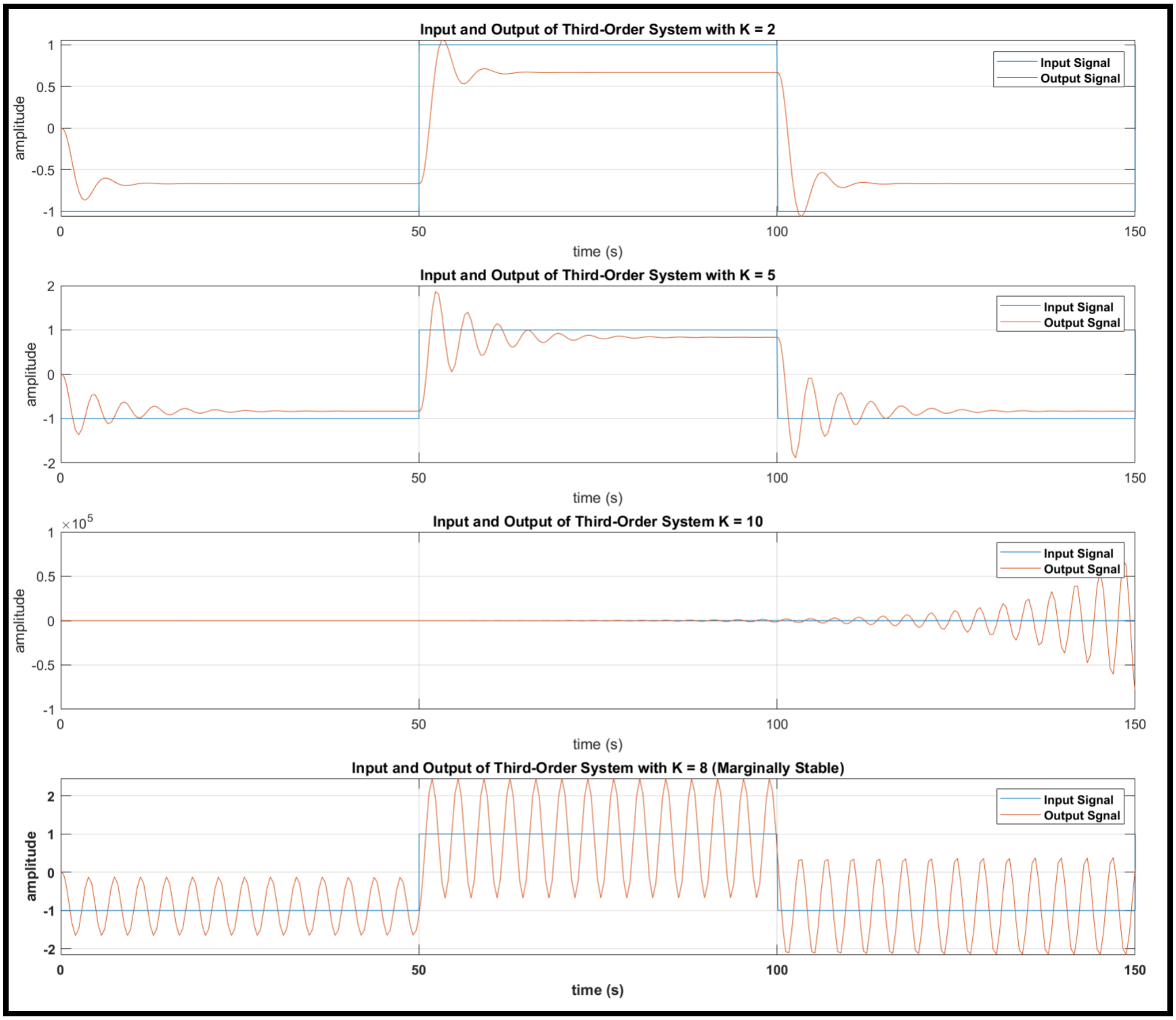


Fig 59: Input and Output of the Closed-Loop System

There is a limitation on the increasing again K in terms of stability which is clearly shown in the calculations and in figure 59 at a maximum value of 8. Any gain past that point moves the poles to the positive side of the imaginary axis making the output unstable.

Summaries

Tanvir Hassan *501104056*

This lab focuses on closed-loop and open-loop control systems, transient responses, and system stability. The initial segment introduces essential concepts and skills to utilize Simulink, first-order systems, and closed-loop and open-loop systems. The latter segment involves the practical application of closed-loop and open-loop systems in analyzing the time response of higher-order systems and the impact of variable gains on these systems.

The lab begins with exploring the MATLAB tool Simulink by making a sinusoidal function. This involves generating the function, its derivative, and its integral and then combining these signals using a multiplexer to then transfer the data into MATLAB workspace. This process is repeated with a step function and a signal generator to accommodate various function inputs. The functions are linked to transfer function blocks, producing outputs as demonstrated in B.1. then A.4 employs three different first-order transfer functions to show the significance of poles and zeros, highlighting that positive poles lead to output divergence, whereas negative poles ensure convergence.

Furthermore, emphasizing the first-order system properties, mainly focusing on determining the DC gain and time constant. [The time constant is identified when the output reaches 63% of its steady-state amplitude]. Understanding the DC gain and time constant is crucial for the rest of the lab, as it aids in interpreting the graphical representations of transfer functions. The behaviors of closed-loop and open-loop control systems are also explored, showing similar responses.

Part 2 of the lab report focuses on examining higher-order systems, mainly how changes influence their time responses, highlighting the delay in achieving steady state. It explores the effects of increasing gain on system characteristics such as overshoot and steady-state error. It was observed that increasing the gain resulted in a rise in %Max Overshoot however the steady-state error remained relatively consistent across both systems, except the case of K=10 third-degree polynomial. The report demonstrates the relationship between gain adjustments and the dynamic behavior of second and third-order systems, highlighting the critical balance between achieving desired performance and maintaining system stability.

We focused on understanding the concepts of open-loop and closed-loop control systems, along with their transient responses and stability characteristics. The first part was an introduction that provided an overview of using Simulink, a MATLAB tool, for modelling first-order systems, and differentiating between open-loop and closed-loop configurations.

Part 1 of the lab introduced Simulink through the development of a sinusoidal function. The function, its derivative, and its integral then had their data transferred into the MATLAB workspace using the workspace block which connected all three signals together through the use of a multiplexer. Subsequently, these functions were linked to transfer function blocks to generate varied outputs, as demonstrated in section B.1. It illustrated the effect of the system poles and zeros on first-order transfer functions. The analysis highlighted that a positive pole in a first-order system leads to divergent output, whereas a negative pole ensures convergence.

The next part determined the importance of DC gain and the time constant. The time constant was identified by measuring the output's amplitude when it reached 63% of its steady-state value. These parameters were used to interpret the transfer functions' graphical representations and understanding the system's behaviour under various conditions. This part of the lab emphasized how changes in poles or gains influence the system's time constant and steady-state error differently in open-loop versus closed-loop systems.

Part two of the lab looked more into higher-order systems with part A showing the delay that corresponds to higher-order systems when trying to reach a steady state. This led to a detailed examination of both second and third-order systems to assess their time domain specifications and the impact of gain adjustments. It was observed that increasing the gain resulted in a higher percentage of maximum overshoot, while the steady-state error remained relatively constant, except for the instability encountered in the system with a gain setting of k = 10.

Appendix

```
%% A.2
SignalInput=out.simoutA1(:,1);
IntegratedSignal=out.simoutA1(:,2);
DerivativeSignal=out.simoutA1(:,3);
subplot(3,1,1);
plot(out.tout, SignalInput , 'Color', '#637000', 'lineWidth', 1.25, 'DisplayName', "x_(t) = 
sin(t)");
title("Original Sine Wave Signal");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
subplot(3,1,2);
plot(out.tout, IntegratedSignal , 'Color', '#487F9C', 'lineWidth', 1.25, 'DisplayName', 
"X_(t) = \integral sin(t)");
title("Integrated Signal");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
subplot(3,1,3);
plot(out.tout, DerivativeSignal , 'Color', '#DB435E', 'lineWidth', 1.25, 'DisplayName', 
"x_(t)' = d/dt (sin(t))");
title("Derived Signal");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
```

```
%% A.3
SignalInput=out.simoutA3(:,1);
IntegratedSignal=out.simoutA3(:,2);
DerivativeSignal=out.simoutA3(:,3);
subplot(3,1,1);
plot(out.tout, SignalInput , 'Color', '#637000', 'lineWidth', 1.25, 'DisplayName', "x_(t) = 
sin(t)");
title("Original Sine Wave Signal");
xlabel("Time");
ylabel("Amplitude");
grid;
subplot(3,1,2);
plot(out.tout, IntegratedSignal  , 'Color', '#487F9C', 'lineWidth', 1.25, 'DisplayName', 
"X_(t) = \integral sin(t)");
title("Integrated Signal");
xlabel("Time");
ylabel("Amplitude");
grid;
subplot(3,1,3);
plot(out.tout, DerivativeSignal , 'Color', '#DB435E', 'lineWidth', 1.25, 'DisplayName', 
"x_(t)' = d/dt (sin(t))");
title("Derived Signal");
xlabel("Time");
ylabel("Amplitude");
grid;
```

```
%% A.4
SignalInput=out.simoutA4(:,1);
T1=out.simoutA4(:,2);
T2=out.simoutA4(:,3);
T3=out.simoutA4(:,4);
T4=out.simoutA4(:,5);
figure
tiledlayout(4,1)
nexttile
plot(out.tout, SignalInput, 'Color', '#487F9C', 'lineWidth', 2.25, 'DisplayName', 'Unit
```

```
Step'), grid
hold on
plot(out.tout, T1, 'Color', '#DB435E', 'lineWidth', 2.25, 'DisplayName', 'T1 Signal')
title('Transfer Function Signal 1', 'FontSize',12);
ylabel('Magnitude');
legend
hold off
nexttile
plot(out.tout, SignalInput, 'Color', '#487F9C', 'lineWidth', 2.25, 'DisplayName', 'Unit
Step'), grid
hold on
plot(out.tout, T2, 'Color', '#DB435E', 'lineWidth', 2.25, 'DisplayName', 'T2 Signal')
title('Transfer Function Signal 2', 'FontSize',12);
ylabel('Magnitude');
legend
hold off
nexttile
plot(out.tout, SignalInput, 'Color', '#487F9C', 'lineWidth', 2.25, 'DisplayName', 'Unit
Step'), grid
hold on
plot(out.tout, T3, 'Color', '#DB435E', 'lineWidth', 2.25, 'DisplayName', 'T3 Signal')
title('Transfer Function Signal 3', 'FontSize',12);
ylabel('Magnitude');
legend
hold off
nexttile
plot(out.tout, SignalInput, 'Color', '#487F9C', 'lineWidth', 2.25, 'DisplayName', 'Unit
Step'), grid
hold on
plot(out.tout, T4, 'Color', '#DB435E', 'lineWidth', 2.25, 'DisplayName', 'T4 Signal')
title('Transfer Function Signal 4', 'FontSize',12);
ylabel('Magnitude');
xlabel('Time (t)');
legend
hold off
```

```
%% B.1
% time = out.tout(:,1);
% step = out.Fstep(:,1);
% response = out.Fstep(:,2);
figure
hold on
plot(out.Fstep , 'lineWidth', 2.25);
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Function & First Order System', 'FontSize',12);
legend ('Step Function', 'First Order System');
grid on;
ylim([-0.1 1.1]);
hold off
```

```
%% C.1
figure
hold on
plot(out.time(:,1),out.FOLstep(:,1) , 'lineWidth', 2.25);
plot(out.time(:,1),out.FOLstep(:,2) , 'lineWidth', 2.25);
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Function & First Order System', 'FontSize',12);
legend ('Step Function', 'First Order System');
grid on;
dcm = datacursormode;
dcm.Enable = 'on';
dcm.DisplayStyle = 'window';
```

```
hold off
```

```
%% C.2
figure
hold on
plot(out.time(:,1),out.FCLstep(:,1) , 'lineWidth', 2.25);
plot(out.time(:,1),out.FCLstep(:,2) , 'lineWidth', 2.25);
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Function & First Order System Closed Loop', 'FontSize',12);
legend ('Step Function', 'First Order System');
grid on;
hold off
```

```
%% C.4
figure
hold on
subplot(3,1,1);
plot(out.time(:,1),out.FOLstepControl(:,1) , 'lineWidth', 2.25);
title("Control Signal u(t)");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
subplot(3,1,2);
plot(out.time(:,1),out.FOLstepControl(:,2) , 'lineWidth', 2.25);
title("First Order System Open Loop");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
subplot(3,1,3);
plot(out.time(:,1),out.FCLstepControl(:,2) , 'lineWidth', 2.25);
title("First Order System Closed Loop");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
% dcm = datacursormode;
% dcm.Enable = 'on';
% dcm.DisplayStyle = 'window';
hold off
```

```
%% C.51
figure
hold on
subplot(4,1,1);
plot(out.time(:,1),out.FOLstepC51(:,1) , 'lineWidth', 2.25);
title("Open Loop Input");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
subplot(4,1,2);
plot(out.time(:,1),out.FOLstepC51(:,2) , 'lineWidth', 2.25);
title("First Order System Open Loop");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
subplot(4,1,3);
plot(out.time(:,1),out.FCLstepC51(:,1) , 'lineWidth', 2.25);
title("Closed Loop Input");
xlabel("Time (s)");
ylabel("Amplitude");
xlim([0 10]);
grid;
subplot(4,1,4);
plot(out.time(:,1),out.FCLstepC51(:,2) , 'lineWidth', 2.25);
```

```matlab
title("First Order System Closed Loop");
xlabel("Time (s)");
ylabel("Amplitude");
xlim([0 10])
grid;

hold off
```

```matlab
%% C.52
figure
hold on
subplot(4,1,1);
plot(out.time(:,1),out.FOLstepC52(:,1) , 'lineWidth', 2.25);
title("Open Loop Input");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
subplot(4,1,2);
plot(out.time(:,1),out.FOLstepC52(:,2) , 'lineWidth', 2.25);
title("First Order System Open Loop");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
subplot(4,1,3);
plot(out.time(:,1),out.FCLstepC52(:,1) , 'lineWidth', 2.25);
title("Closed Loop Input");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
subplot(4,1,4);
plot(out.time(:,1),out.FCLstepC52(:,2) , 'lineWidth', 2.25);
title("First Order System Closed Loop");
xlabel("Time (s)");
ylabel("Amplitude");
grid;
```

```matlab
%% A1
figure
hold on
plot(out.time(:,1),out.squareTran(:,1) , 'lineWidth', 2.25);
plot(out.time(:,1),out.squareTran(:,2) , 'lineWidth', 2.25);
plot(out.time(:,1),out.squareTran(:,3) , 'lineWidth', 2.25);
plot(out.time(:,1),out.squareTran(:,4) , 'lineWidth', 2.25);
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Function & High Order Systems', 'FontSize',12);
legend ('Step Function', 'First Order System', 'Second Order System', 'Third Order System');
grid on;
hold off
```

```matlab
%% B1
figure
hold on
plot(out.time(:,1),out.SOstep(:,1) , 'lineWidth', 2.25);
plot(out.time(:,1),out.SOstep(:,2) , 'lineWidth', 2.25);
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Function & Second Order System', 'FontSize',12);
legend ('Step Function', 'Second Order System');
grid on;
hold off
```

```matlab
%% B2
k2 = [1 2 3];
```

```
K2 = roots (k2)
k5 = [1 2 6];
K5 = roots (k5)
k10 = [1 2 11];
K10 = roots (k10)
```

```
%% C1
figure
hold on
plot(out.time(:,1),out.TOstepK2(:,1) , 'lineWidth', 2.25);
plot(out.time(:,1),out.TOstepK2(:,2) , 'lineWidth', 2.25);
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Function & Third Order System k=2', 'FontSize',12);
legend ('Step Function', 'Third Order System');
grid on;
hold off
figure
hold on
plot(out.time(:,1),out.TOstepK5(:,1) , 'lineWidth', 2.25);
plot(out.time(:,1),out.TOstepK5(:,2) , 'lineWidth', 2.25);
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Function & Third Order System k=5', 'FontSize',12);
legend ('Step Function', 'Third Order System');
grid on;
hold off
figure
hold on
plot(out.time(:,1),out.TOstepK10(:,1) , 'lineWidth', 2.25);
plot(out.time(:,1),out.TOstepK10(:,2) , 'lineWidth', 2.25);
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Function & Third Order System k=10', 'FontSize',12);
legend ('Step Function', 'Third Order System');
grid on;
hold off
```

```
%% C2
k2 = [1 3 3 3];
K2 = roots (k2)
k5 = [1 3 3 6];
K5 = roots (k5)
k10 = [1 3 3 11];
K10 = roots (k10)
```

```
%% C3
num2 = 2;
k2 = [1 3 3 3];
sys1 = tf(num2,k2);
stepinfo(sys1)
stepplot(sys1)
%%
num5 = 5;
k5 = [1 3 3 6];
sys2 = tf(num5,k5);
stepinfo(sys2)
stepplot(sys2)
%%
num10 = 10;
k10 = [1 3 3 11];
sys3 = tf(num10,k10);
stepinfo(sys3)
stepplot(sys3)
```

```
%% C4
hold on
subplot(4,1,1)
plot(out.time,out.TOstepK2(:,1:2));
xlabel('time (s)')
ylabel('amplitude')
title('Input and Output of Third-Order System with K = 2')
grid on
subplot(4,1,2)
plot(out.time,out.TOstepK5(:,1:2));
xlabel('time (s)')
ylabel('amplitude')
title('Input and Output of Third-Order System with K = 5')
grid on
subplot(4,1,3)
plot(out.time,out.TOstepK10(:,1:2));
xlabel('time (s)')
ylabel('amplitude')
title('Input and Output of Third-Order System K = 10')
grid on
subplot(4,1,4)
plot(out.time,out.TOstep8(:,1:2));
xlabel('time (s)')
ylabel('amplitude')
title('Input and Output of Third-Order System with K = 8 (Marginally Stable)')
grid on
hold off
```