Faculty of Enigeering

Cairo University

Modern Control Project

Bode Plot & State Space Representation

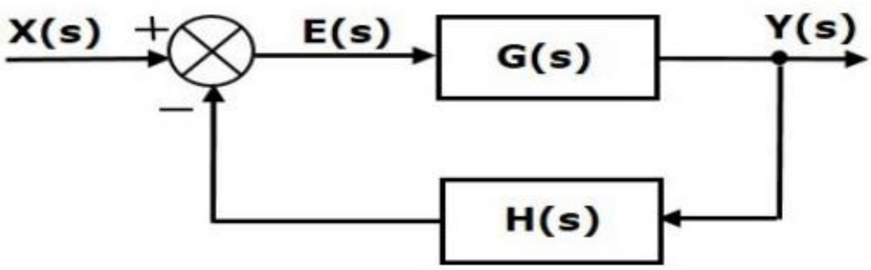
Presented for **ELC 3090** MATLAB Project

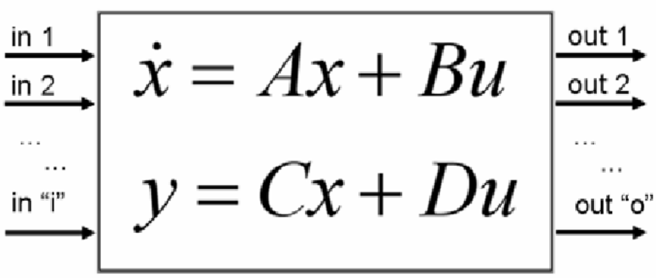
Presented to:

Dr. Mina & Hannan

T.A: Mohamed Mostafa

(3^{rd} Year Electronics and Electrical Communication Engineers)	
مجدي أحمد عباس عبد الحميد الابرق	Sec: 3 / I.D: 9210899 / BN: 36





Task 1

Control Project

Bode Plot

[1] G(s) & H(s) Definition using tf command

```
G =

     1
  -------
  s^2 + s

Continuous-time transfer function.


H =

  1

Static gain.
```

```
%% Bode Plot
%% [1] G(s) & H(s) Definition using tf command
num_G = [0 0 1];
denum_G = [1 1 0];
num_H = [0 0 1];
denum_H = [0 0 1];
G = tf(num_G, denum_G);
H = tf(num_H, denum_H);
```

[2] Plot G(s) Response for a Unit Step Input

```
%% [2] Plot G(s) Response for a Unit Step Input
figure;
step(G);
title('Step Response of G(s)');
grid on;
% Unstable Response + Expected Behavior (poles = 0 ,−1)
poles_G = eig(G); % = pole(G);
if (real(poles_G(:)) < 0)
  disp('G(s) Output Due to Unit Step input is Stable');
elseif (real(poles_G(:)) == 0)
  disp('G(s) Output Due to Unit Step input is Critically Stable');
else
  disp('G(s) Output Due to Unit Step input is Unstable');
end
```

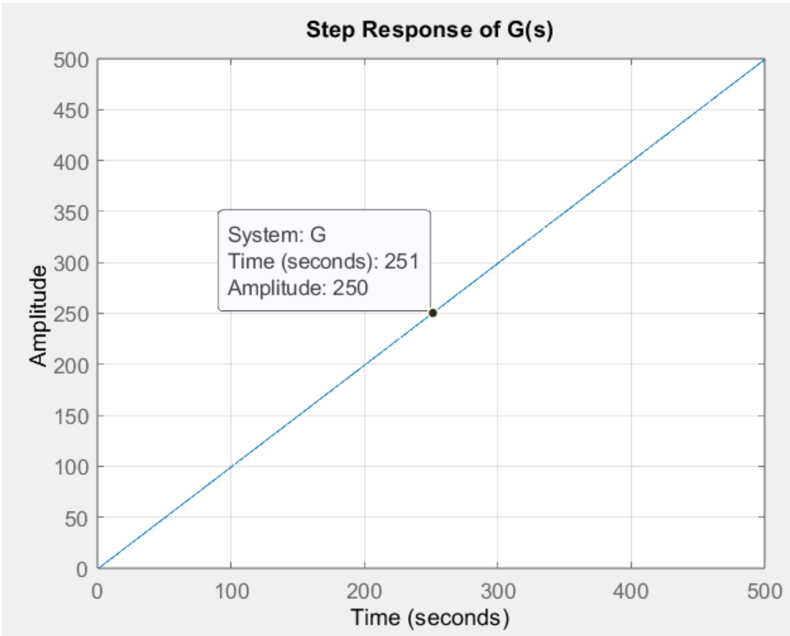


Figure 1: Step Response of G(s)

```
poles_G =

     0
    -1

G(s) Output Due to Unit Step input is Unstable
```

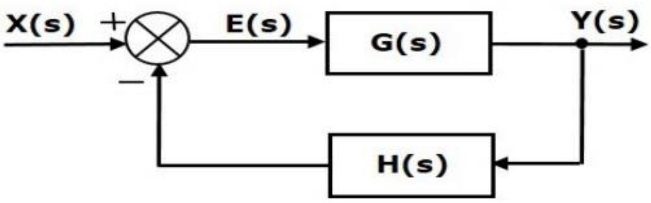
Comments: -

The plot generated by **step(G)** will show the **G** output response over time when a unit step input is applied. In this case, the output is **Unstable**, you'll likely observe unbounded growth in the output signal, indicating the instability as shown in Figure (1). This behavior aligns with our expectation that the output has pole at $(s = 0)$ will exhibit **Unstable** Integrator (Accumulator) so it will go to infinity and because the denominator of G(s) is a polynomial with zero constant coefficient (Zero term free of S), so it isn't saturate as expected.

[3] Closed Loop Feedback System

Hand Analysis: -

Block Diagram of Closed loop System



$X(s)$: Reference Input Signal

$E(s)$: Actuating or Error Signal

$G(s)$: Feedforward Transfer Function

$H(s)$: Feedback Transfer Function

$Y(s)$: Output Signal

Closed Loop Transfer function Hand Analysis for Negative Feedback System

$Y(s) = G(s).E(s) \rightarrow (1)$

$E(s) = X(s) - Y(s).H(s) \rightarrow (2)$

Substitute (2) in (1):

$Y(s) = G(s)[X(s) - Y(s).H(s)]$

$Y(s) + Y(s).G(s).H(s) = G(s).X(s)$

$$\frac{Y(s)}{X(s)} = \frac{G(s)}{1 + G(s).H(s)} = \frac{\frac{1}{s(s+1)}}{1 + \frac{1}{s(s+1)}} \times \frac{s(s+1)}{s(s+1)} = \frac{1}{s(s+1)+1} = \frac{1}{s^2+s+1} \qquad \boxed{1}$$

```
%% [3] Closed Loop Feedback System Definition
CL_F = feedback(G,H,−1); % −1 = −ve feedback
CL_Formula = G/(1 + G ∗ H); % Closed Loop Formula with unsimplified form
CL_Formula_Simplified = minreal(CL_Formula); % Closed Loop Formula with simplified form
% Comparison between Closed loop system in feedback command and with G/(1 + GH) formula using bode plot
figure;
bode(CL_F,'r');
hold on;
bode(CL_Formula,'b−−o');
hold off;
legend('CL − Feedback',' CL − Formula');
title('Bode Plot Comparison for Closed Loop Transfer Function');
grid on;
```

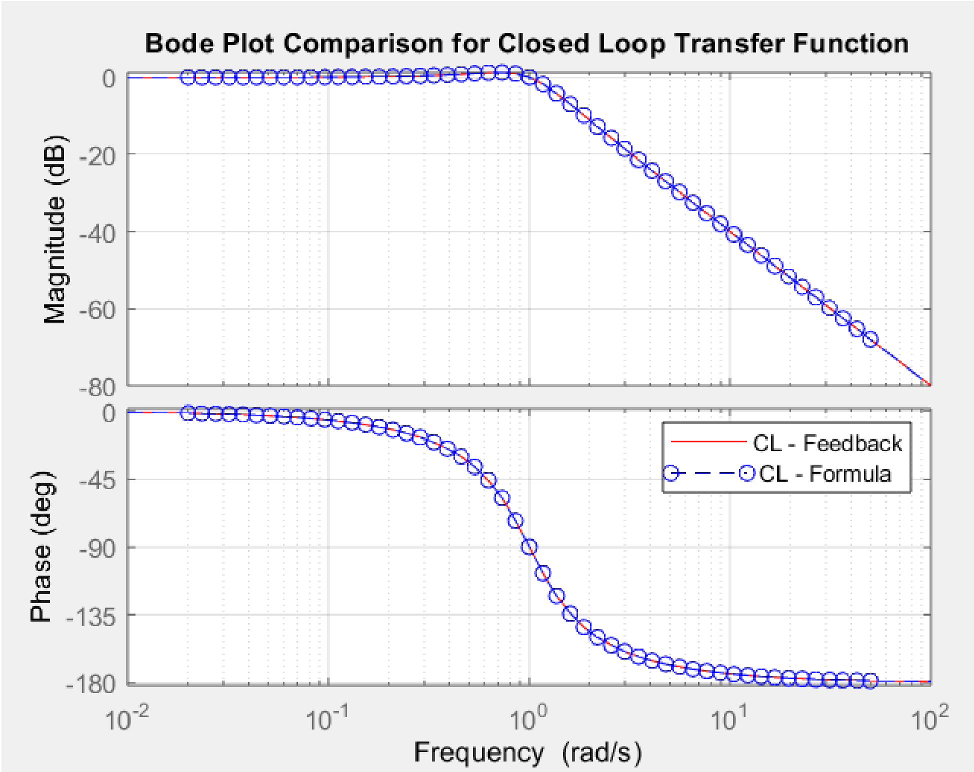


Figure 2: Bode Plot Comparison for Closed Loop Transfer Function using Feedback Command and Formula

```
CL_F =

       1
  -----------
  s^2 + s + 1

Continuous-time transfer function.


CL_Formula =

          s^2 + s
  -----------------------
  s^4 + 2 s^3 + 2 s^2 + s

Continuous-time transfer function.


CL_Formula_Simplified =

       1
  -----------
  s^2 + s + 1

Continuous-time transfer function.
```

2

From [1] & [2]: -

Hand Analysis and Simulated Transfer Functions are the Same.

[4] Plot Output of Closed Loop

```
%% [4] Plot Output of Closed Loop
figure;
step(CL_F);
title('Step Response of Closed Loop Transfer Function');
grid on;
poles_CL = eig(CL_F); % = pole(G)
if (real(poles_CL(:)) < 0)
   disp('Closed Loop System Output Due to Unit Step input is Stable');
elseif (real(poles_CL(:)) == 0)
   disp('Closed Loop System Output Due to Unit Step input is Critically Stable');
else
   disp('Closed Loop System Output Due to Unit Step input is Unstable');
end
```

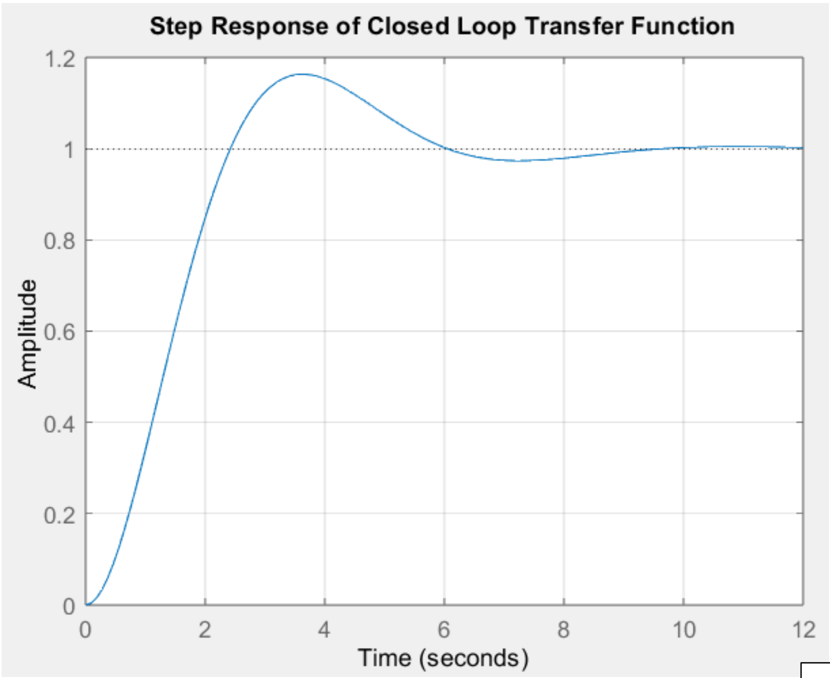


Figure 3: Step Response of Closed Loop Transfer Function

```
poles_CL =

  -0.5000 + 0.8660i
  -0.5000 - 0.8660i

Closed Loop System Output Due to Unit Step input is Stable
```

$c/c: S^2 + S + 1 = 0$

S^2	**1**	**1**
S^1	**1**	**0**
S^0	**1**	

No Sign Changes

All Coeff. Are $+ve.$

Poles in LHP ∴ Stable

Comments: -

The plot generated by **step(G)** will show the **CL** output response over time when a unit step input is applied. In this case, the output is **Stable**, you'll likely observe **Some Damped Oscillations** in the output signal, then Steady State Response as shown in Figure (3).
This behavior aligns with our expectation that the output is **Stable** Using Routh Stability Check.

[5] Plot Output of Closed Loop

Hand Analysis: -

Characteristic Equation: $\boldsymbol{S^2 + S + 1 = 0} \quad \rightarrow \quad \boldsymbol{S_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-1 \pm \sqrt{1-4}}{2} = -0.5 \pm 0.866\, j}$ 1

2

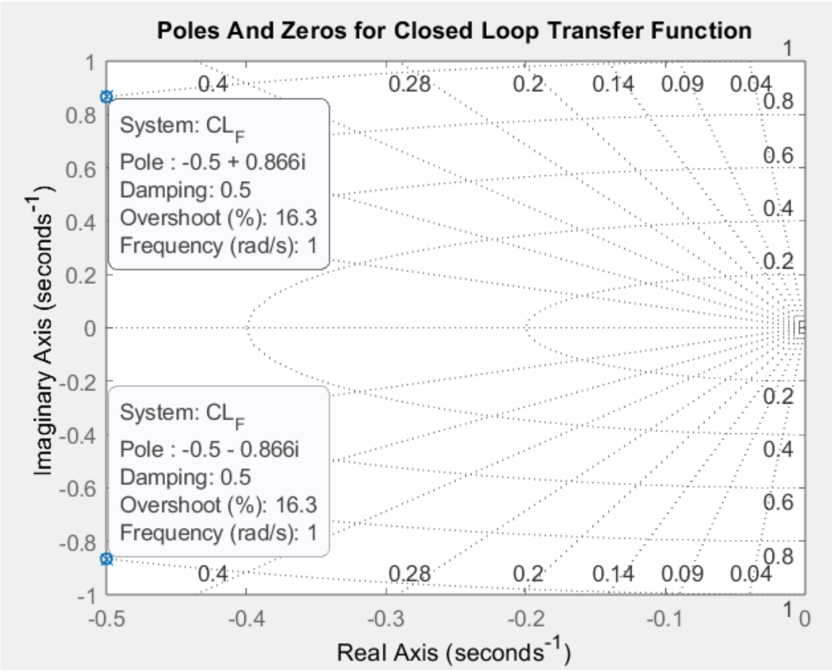


Figure 4: Poles and Zeros for Closed Loop Transfer Function

There are 2 Complex Conjugate Poles (in LHP) and No Zeros so, the system is Stable.

And This Result Agrees with the Results in [4] as $S_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$

From Figure (4): $\omega_n = 1\, rad/sec$, $\zeta\omega_n = \frac{1}{2} \rightarrow \zeta = \frac{1}{2} \rightarrow 0 < \zeta < 1 \rightarrow \therefore$ Underdamped System.

∴ "There are Some Damped Oscillations".

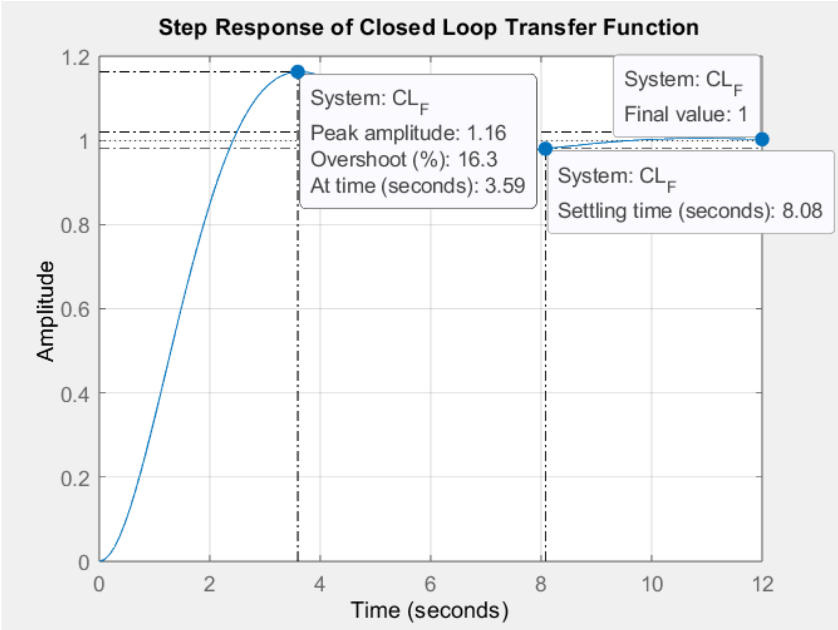
```
%% [5] Poles Locations of the Closed Loop Transfer Function
figure;
pzmap(CL_F);
title('Poles And Zeros for Closed Loop Transfer Function');
grid on;
```

From [1] & [2]: -

Hand Analysis and Simulated Poles are the Same.

[6] Closed Loop Step Response Characteristics

```
%% [6] Closed Loop Step Response Characteristics
% Peak Amplitude  =  1.16, Overshoot  =  16.3 % @ Time  =  3.59 Sec
% Settling Time  =  8.08 Sec
% Steady State Final Value  =  1
```



1

Figure 5: CL TF Step Response Showing Ts, OS%, S.S.

Hand Analysis: -

Overshoot: $OS(\%) = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \times 100\ \% = e^{-\frac{\pi(0.5)}{\sqrt{1-(0.5)^2}}} \times 100\ \% = 16.3\ \%$

Steady State Final Value: $S.S = \lim_{S\to 0} S.TF(S).\frac{1}{S} = \lim_{S\to 0} \frac{1}{S^2+S+1} = 1$

Peak Amplitude:

$$OS = \frac{Peak\ Amplitude - Final\ Value}{Final\ Value} = \frac{16.3}{100} \rightarrow Peak\ Amplitude = OS + 1 = 1.163$$

Peak Time: $T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = \frac{\pi}{1\times\sqrt{1-(0.5)^2}} = 3.63\ sec$

2

Settling Time: $T_s = \frac{4}{\zeta\omega_n} = \frac{4}{0.5} = 8\ sec$

From [1] & [2]: -

Hand Analysis and Simulated 2nd Order Parameters are the Same with small errors.

[7] Steady State Error to a Unit Step Input

```
%% [7] Steady State Error to a Unit Step Input
% ess  =  0  +  System type 1
```

Hand Analysis: -

Open Loop Transfer Function: $G(s).H(s) = \frac{1}{S(S+1)} \rightarrow N = 1\ , Type = 1$ (1st Order Poles)

$X(s) = \frac{1}{S}\ , e_{ss} = \lim_{S\to 0} S.E(s) = \lim_{S\to 0} S.\frac{X(s)}{1+G(s)\ .\ H(s)} = \lim_{S\to 0} S.\frac{\frac{1}{S}}{1+\frac{1}{S^2+S}} = \lim_{S\to 0} \frac{S^2+S}{S^2+S+1} = 0$

This Steady State Error Agrees with type (1) of the System as for type (1): $e_{ss} = 0$

[8] Ramp Input Response

```
%% [8] Ramp Input Response
integrator = tf([0 1], [1 0]);
figure;
step(CL_F.* integrator, 'r');
hold on;
step(integrator, 'b--o');
hold off;
legend('CL TF Ramp Response',' Ramp Input Response');
title('Ramp Response Comparison for Input and Closed Loop Transfer Function');
grid on;
% ess = 1
```

```
integrator =

  1
  -
  s
```

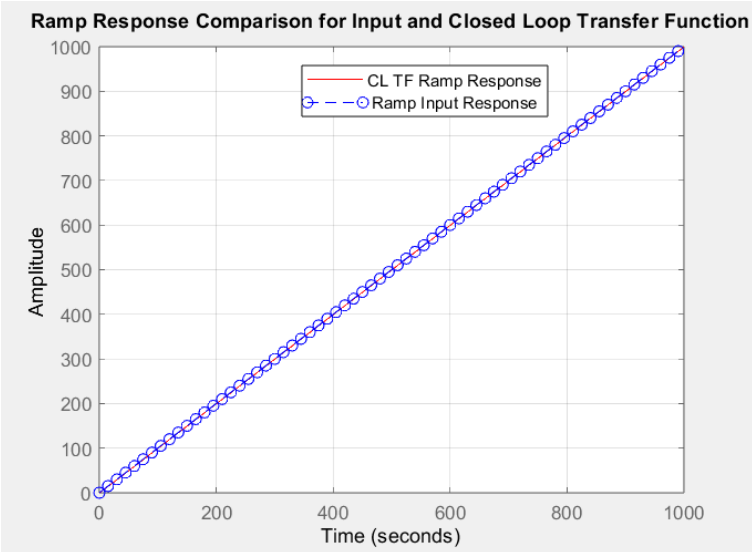


Figure 6: Ramp Response of Closed Loop TF and Input

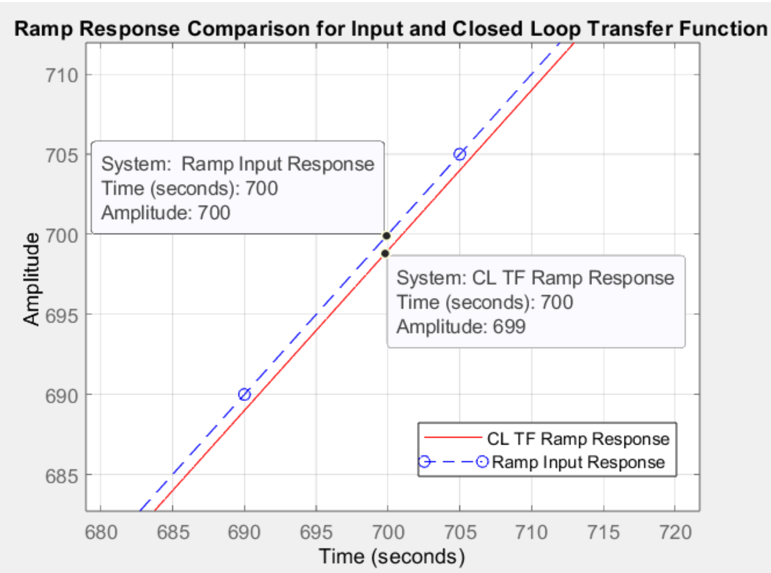


Figure 7: Ramp Response of Closed Loop TF and Input (Zoomed Version)

From Figure 7: $e_{ss_{ramp}} = 700 - 699 = 1$ $\boxed{1}$

Hand Analysis: -

2

$$e_{ss_{ramp}} = \frac{1}{K_v}$$

$$K_v = \lim_{S\to 0} S.G(s).H(s) = \lim_{S\to 0} \frac{S}{S^2+S} = \lim_{S\to 0} \frac{S}{S(S+1)} = \lim_{S\to 0} \frac{1}{(S+1)} = 1$$

$$e_{ss_{ramp}} = \lim_{K_v\to 1} \frac{1}{K_v} = 1$$

From [1] & [2]: -

Hand Analysis and Simulated Steady state Error are the Same.

[9] Plot Frequency Response of the System

```
%% [9] Plot Frequency Response of the System
% ωpc, ωgc, PM, GM are got from Open loop Gain G(s).H(s)
figure;
margin(G * H);
grid on;
allmargin(G * H)
```

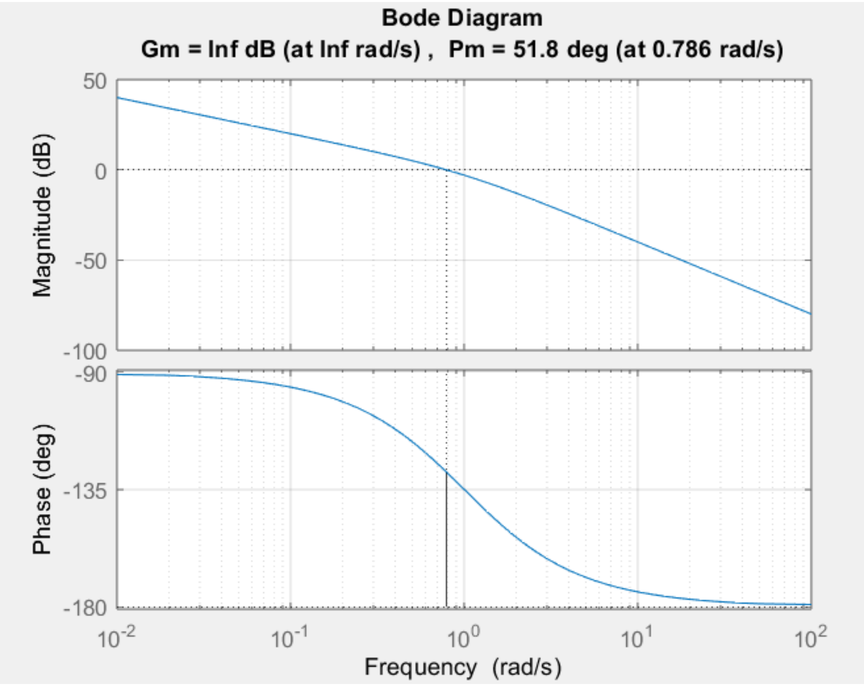


Figure 8: Frequency Response Bode Plot of the System

```
 GainMargin: Inf
GMFrequency: Inf
PhaseMargin: 51.8273
PMFrequency: 0.7862
DelayMargin: 1.1506
DMFrequency: 0.7862
     Stable: 1
```

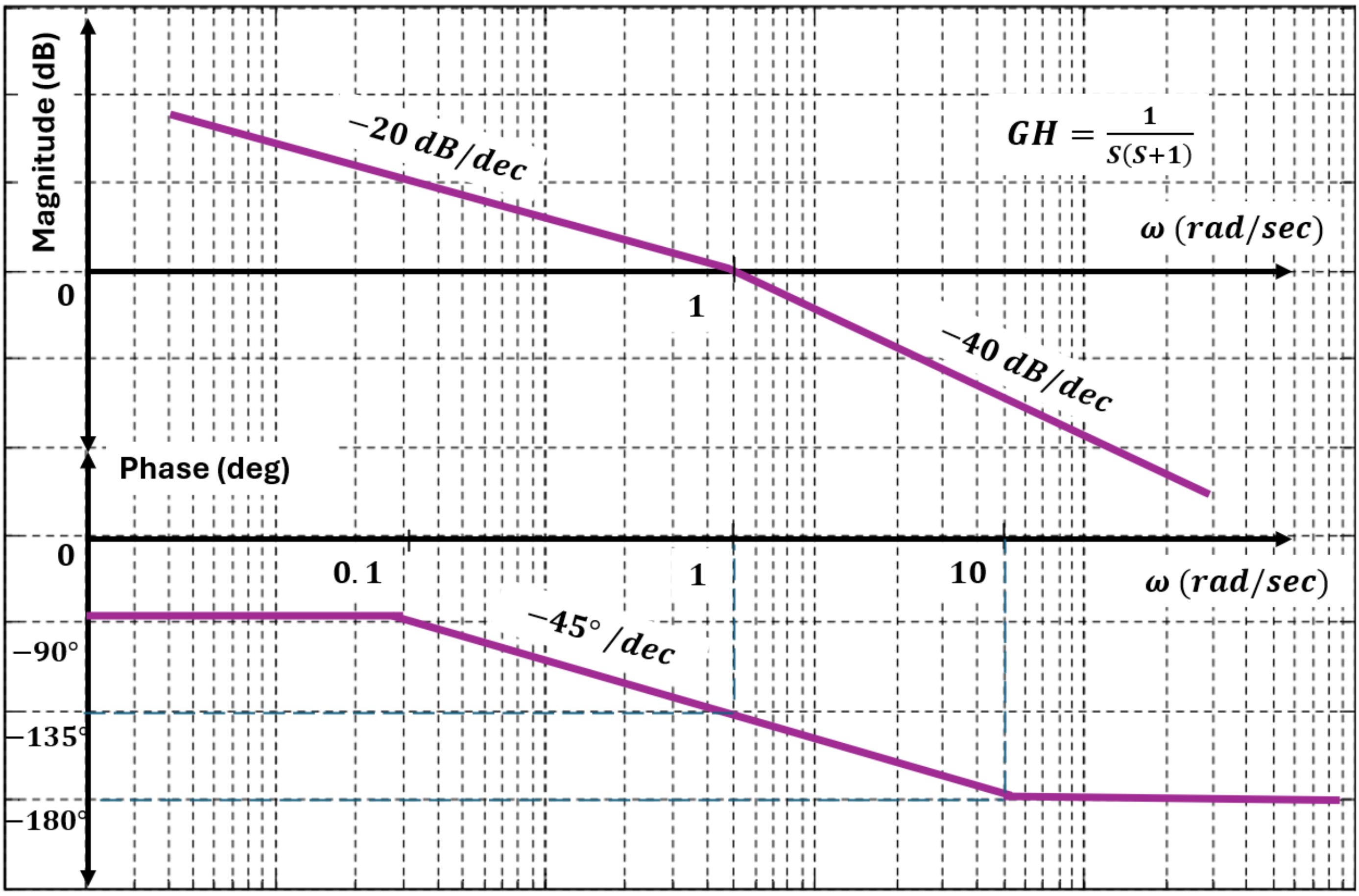


Figure 9: Bode Plot Diagram (Handmade) in PowerPoint ❤

$\omega_{gc} = 1\,rad/sec$

$\angle\,\omega_{gc} = -90° - 45\log\left(\frac{1}{0.1}\right) = -135°$

$PM = 180° + \angle\,\omega_{gc} = 180° - 135° = 45°$

$\omega_{pc} = \infty\,rad/sec$

$GM = -(|mag_{dB}|)_{\omega_{pc}} = -(-\infty) = \infty$

$\because\ \omega_{pc} > \omega_{gc}\ \&\ PM, GM > 0$ $\therefore$ The System is Stable.

P.O.C	MATLAB	Hand Analysis
$\boldsymbol{\omega_{gc}}$	$0.7862\,rad/sec$	$1\,rad/sec$
$\boldsymbol{\angle\,\omega_{gc}}$	$-128°$	$-135°$
$\boldsymbol{PM}$	$51.8273°$	$45°$
$\boldsymbol{\omega_{pc}}$	$\infty\,rad/sec$	$\infty\,rad/sec$
$\boldsymbol{GM}$	$\infty\,dB$	$\infty\,dB$
Comments	MATLAB Results are more accurate than Hand Analysis Results as we make Approximations and Simplifications to make the Sketching easer using Asymptotic Lines not the exact curves leading to differences from Simulation Results	

Bode Plot Full MATLAB Code

```
close all;
clear;
clc;
%% Bode Plot
%% [1] G(s) & H(s) Definition using tf command
num_G = [0 0 1];
denum_G = [1 1 0];
num_H = [0 0 1];
denum_H = [0 0 1];
G = tf(num_G, denum_G);
H = tf(num_H, denum_H);
%% [2] Plot G(s) Response for a Unit Step Input
figure;
step(G);
title('Step Response of G(s)');
grid on;
% Unstable Response + Expected Behavior (poles = 0 , -1)
poles_G = eig(G); % = pole(G);
if (real(poles_G(:)) < 0)
   disp('G(s) Output Due to Unit Step input is Stable');
elseif (real(poles_G(:)) == 0)
   disp('G(s) Output Due to Unit Step input is Critically Stable');
else
   disp('G(s) Output Due to Unit Step input is Unstable');
end
%% [3] Closed Loop Feedback System Definition
CL_F = feedback(G, H, -1); % - 1 = -ve feedback
CL_Formula = G/(1 + G * H); % Closed Loop Formula with unsimplified form
CL_Formula_Simplified
            = minreal(CL_Formula); % Closed Loop Formula with simplified form
% Comparison between Closed loop system in feedback command and with G/(1 + GH)
                                                   formula using bode plot
figure;
bode(CL_F, 'r');
hold on;
bode(CL_Formula, 'b--o');
hold off;
legend('CL - Feedback', ' CL - Formula');
title('Bode Plot Comparison for Closed Loop Transfer Function');
grid on;
%% [4] Plot Output of Closed Loop
figure;
step(CL_F);
title('Step Response of Closed Loop Transfer Function');
```

```
grid on;
poles_CL = eig(CL_F); % = pole(G)
if (real(poles_CL(:)) < 0)
   disp('Closed Loop System Output Due to Unit Step input is Stable');
elseif (real(poles_CL(:)) == 0)
   disp('Closed Loop System Output Due to Unit Step input is Critically Stable');
else
   disp('Closed Loop System Output Due to Unit Step input is Unstable');
end
%% [5] Poles Locations of the Closed Loop Transfer Function
figure;
pzmap(CL_F);
title('Poles And Zeros for Closed Loop Transfer Function');
grid on;
%% [6] Closed Loop Step Response Characteristics
% Peak Amplitude = 1.16, Overshoot = 16.3 % @ Time = 3.59 Sec
% Settling Time = 8.08 Sec
% Steady State Final Value = 1
%% [7] Steady State Error to a Unit Step Input
% ess = 0 + System type 1
%% [8] Ramp Input Response
integrator = tf([0 1], [1 0]);
figure;
step(CL_F.* integrator, 'r');
hold on;
step(integrator, 'b--o');
hold off;
legend('CL TF Ramp Response', ' Ramp Input Response');
title('Ramp Response Comparison for Input and Closed Loop Transfer Function');
grid on;
% ess = 1
%% [9] Plot Frequency Response of the System
% ωpc, ωgc, PM, GM are got from Open loop Gain G(s). H(s)
figure;
margin(G * H);
grid on;
allmargin(G * H)
```

Task 2

State Space Representation

[1] Finding Transfer Function Hand

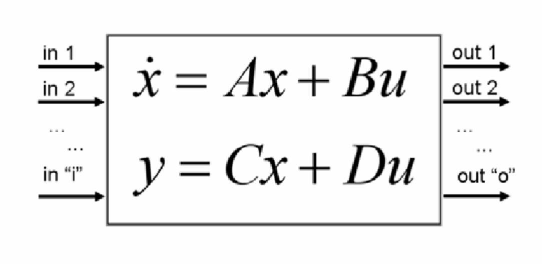
$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -6 & -5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)\,,\ y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

$A = \begin{bmatrix} 0 & 1 \\ -6 & -5 \end{bmatrix}$ ≡ System (State) Matrix

$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ ≡ Input Matrix

$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$ ≡ Output Matrix

$D = [0]$ ≡ Feedforward Matrix



Transfer Function:

$\underline{\dot{X}} = \underline{A}\,\underline{X} + \underline{B}\,\underline{U}\,,\ \underline{Y} = \underline{C}\,\underline{X} + \underline{D}\,\underline{U}$ Applying $\mathcal{L}\{\}$ Laplace Transform (S-Domain) with 0-initial conditions

$$\underline{X} = [\,(S\underline{I} - \underline{A})^{-1}\underline{B}\,].\underline{U} \rightarrow \underline{Y} = (\,\underline{C}.[\,(S\underline{I} - \underline{A})^{-1}\underline{B}\,] + \underline{D}\,).\underline{U}$$

$$H(S) = \frac{Y(S)}{U(S)} = \underline{C}.[\,(S\underline{I} - \underline{A})^{-1}\underline{B}\,] + \underline{D}$$

$$S\underline{I} - \underline{A} = \begin{pmatrix} S & 0 \\ 0 & S \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -6 & -5 \end{pmatrix} = \begin{pmatrix} S & -1 \\ 6 & S+5 \end{pmatrix}\,,\ (S\underline{I} - \underline{A})^{-1} = \frac{1}{S^2+5S+6}\begin{pmatrix} S+5 & 1 \\ -6 & S \end{pmatrix}$$

$$H(S) = \frac{1}{S^2+5S+6}\left[\begin{pmatrix} 1 & 0 \end{pmatrix}\begin{pmatrix} S+5 & 1 \\ -6 & S \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right] = \frac{1}{S^2+5S+6}\left[\begin{pmatrix} S+5 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right] = \frac{1}{S^2+5S+6}$$

$$\boxed{\therefore H(S) = \frac{1}{S^2+5S+6}} \qquad \boxed{1}$$

```
%% State Space Representation
%% [1] Finding Transfer Function Hand Analysis
% T.F.= C ((SI – A)^ – 1) B + D
% T.F.= 1 / (s^2 + 5s + 6)
```

[2] Finding Transfer Function From States and Output

```
%% [2] Finding Transfer Function From States and Output Matrices
syms s; % Complex Frequency Domain
A = [0 1 ; −6 − 5]; % A = System (State) Matrix
B = [0 ; 1];      % B = Input Matrix
C = [1 0];        % C = Output Matrix
D = 0;            % D = Feedforward Matrix
system = ss(A, B, C, D);
TF_1 = C ∗ (inv(s .∗ eye(2) − A)) ∗ B + D; % Controllable Form
[num, denum] = ss2tf(A, B, C, D); % ss2tf Command Convert A, B, C, D to TF
TF_2 = tf(num, denum);
```

```
system =

  A =
         x1   x2
   x1     0    1
   x2    -6   -5

  B =
         u1
   x1     0
   x2     1

  C =
         x1   x2
   y1     1    0

  D =
         u1
   y1     0

Continuous-time state-space model.
```

```
TF_1 =

1/(s^2 + 5*s + 6)
```

2

```
num =

     0     0     1


denum =

    1.0000    5.0000    6.0000


TF_2 =

        1
  -------------
  s^2 + 5 s + 6

Continuous-time transfer function.
```

3

From [1], [2] & [3]: -

Hand Analysis, using Inverse and eye Function, and $ss2tf(A, B, C, D)$ Function they are the Same.

[3] State Transition Matrix

```
%% [3] State Transition Matrix
syms t; % Time Domain
PHI_S = inv(s.* eye(2) - A);
PHI_T = ilaplace(PHI_S);
PHI_0 = subs(PHI_T,t,0); % subs(function, old, new) PHI_0 = I
if (unique(PHI_0 == eye(2)))
  disp('Ø(0) = I');
else
  disp('Ø(0) ≠ I');
end
```

```
PHI_S =

[(s + 5)/(s^2 + 5*s + 6), 1/(s^2 + 5*s + 6)]
[     -6/(s^2 + 5*s + 6), s/(s^2 + 5*s + 6)]


PHI_T =

[3*exp(-2*t) - 2*exp(-3*t),     exp(-2*t) - exp(-3*t)]
[6*exp(-3*t) - 6*exp(-2*t), 3*exp(-3*t) - 2*exp(-2*t)]


PHI_0 =

[1, 0]
[0, 1]

Ø(0) = I
```

[4] Derivative of State Transition Matrix

```
%% [4] Derivative of State Transition Matrix
PHI_Diff = diff(PHI_T,t);
APHI = A * PHI_T;
if (unique(PHI_Diff == APHI))
  disp('dØ/dt = A.Ø(t)');
else
  disp('dØ/dt ≠ A.Ø(t)');
end
```

```
PHI_Diff =

[  6*exp(-3*t) - 6*exp(-2*t), 3*exp(-3*t) - 2*exp(-2*t)]
[12*exp(-2*t) - 18*exp(-3*t), 4*exp(-2*t) - 9*exp(-3*t)]


APHI =

[  6*exp(-3*t) - 6*exp(-2*t), 3*exp(-3*t) - 2*exp(-2*t)]
[12*exp(-2*t) - 18*exp(-3*t), 4*exp(-2*t) - 9*exp(-3*t)]

dØ/dt = A.Ø(t)
```

[5] Controllability and Observability of this SSR

```
%% [5] Controllability and Observability of this SSR
% Checking Contrillability Δ(Qc) ≠ 0 (Expected Controllable)
Rank_C = rank(ctrb(ss(system)));
degree = length(denum) − 1;
if (Rank_C == degree)
  disp(['System is Controllable with Rank = ',num2str(degree)]);
else
  disp('System is Uncontrollable');
end
% Checking Observability Δ(Qo) ≠ 0 (Expected Observable)
Rank_O = rank(obsv(ss(system)));
if (Rank_O == degree)
  disp(['System is Observable with Rank = ',num2str(degree)]);
else
  disp('System is Non Observable');
end
```

```
System is Controllable with Rank = 2
System is Observable with Rank = 2
```

Expectations:

$$A = \begin{bmatrix} 0 & 1 \\ -6 & -5 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 \end{bmatrix},\ D = [0]$$

$$[AB] = \begin{pmatrix} 0 & 1 \\ -6 & -5 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -5 \end{pmatrix} \quad , \quad [CA] = \begin{pmatrix} 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ -6 & -5 \end{pmatrix} = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

$$Q_c = [B \quad AB] = \begin{pmatrix} 0 & 1 \\ 1 & -5 \end{pmatrix},\Delta(Q_c) = \begin{vmatrix} 0 & 1 \\ 1 & -5 \end{vmatrix} = -1 \neq 0 \rightarrow Rank(Q_c) = 2 \rightarrow (\text{Controllable})$$

$$Q_o = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\Delta(Q_o) = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = 1 \neq 0 \rightarrow Rank(Q_o) = 2 \rightarrow (\text{Observable})$$

[6] Unforced (Homogeneous) Solution of the States x(t) and The Unforced Response y(t)

```
%% [6] Unforced (Homogeneous) Solution of the States x(t) and The Unforced Response y(t)
x_0 = [0 ; 1];
x_unforced = PHI_T * x_0;
y_unforced = C * x_unforced;
```

```
x_0 =          x_unforced =                      y_unforced =

     0              exp(-2*t) - exp(-3*t)
     1         3*exp(-3*t) - 2*exp(-2*t)         exp(-2*t) - exp(-3*t)
```

[7] The Unhomogeneous System

```
%% [7] The Unhomogeneous System
U = 1/s;
x_forced = x_unforced + ilaplace((inv(s.* eye(2) - A)) * B * U);
y_forced = C * x_forced;
```

```
x_forced =

exp(-2*t)/2 - (2*exp(-3*t))/3 + 1/6
              2*exp(-3*t) - exp(-2*t)


y_forced =

exp(-2*t)/2 - (2*exp(-3*t))/3 + 1/6
```

Expectations:

$$\underline{x}_{\text{unforced}} = \phi(t)\cdot \underline{x}_0(t) = \begin{pmatrix} 3e^{-2t}-2e^{-3t} & e^{-2t}-e^{-3t} \\ -6e^{-2t}+6e^{-3t} & -2e^{-2t}+3e^{-3t} \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} e^{-2t}-e^{-3t} \\ -2e^{-2t}+3e^{-3t} \end{pmatrix}$$

$$Y_{\text{unforced}} = \underline{C}\,\underline{x}_{\text{unforced}} = \begin{pmatrix} 1 & 0 \end{pmatrix}\begin{pmatrix} e^{-2t}-e^{-3t} \\ -2e^{-2t}+3e^{-3t} \end{pmatrix} = \left(e^{-2t}-e^{-3t}\right)$$

$$x_{\text{forced}} = x_{\text{unforced}} + x_{\text{non-homogeneous}}$$

$$x_{\text{non-homogeneous}} = \int_{-\infty}^{t} \phi(\tau)\cdot \underline{B}u(t-\tau) = \int_0^t \phi(t)\cdot \underline{B} = \int_0^t \begin{pmatrix} e^{-2\tau}-e^{-3\tau} \\ -2e^{-2\tau}+3e^{-3\tau} \end{pmatrix} d\tau = \begin{pmatrix} \frac{1}{6}-\frac{1}{2}e^{-2t}+\frac{1}{3}e^{-3t} \\ e^{-2t}-e^{-3t} \end{pmatrix}$$

$$x_{\text{forced}} = \begin{pmatrix} \frac{1}{6}+\frac{1}{2}e^{-2t}-\frac{2}{3}e^{-3t} \\ -e^{-2t}+2e^{-3t} \end{pmatrix}$$

$$y_{\text{forced}} = \underline{C}.\underline{x}_{\text{forced}} = \frac{1}{6}+\frac{1}{2}e^{-2t}-\frac{2}{3}e^{-3t}$$

SSR Full MATLAB Code

```
close all;
clear;
clc;
%% State Space Representation
%% [1] Finding Transfer Function Hand Analysis
% T.F.= C ((SI – A)^ – 1) B + D
% T.F.= 1 / (s^2 + 5s + 6)
%% [2] Finding Transfer Function From States and Output Matrices
syms s; % Complex Frequency Domain
A = [0 1 ; -6 - 5]; % A = System (State) Matrix
B = [0 ; 1];      % B = Input Matrix
C = [1 0];        % C = Output Matrix
D = 0;            % D = Feedforward Matrix
system = ss(A, B, C, D);
TF_1 = C * (inv(s .* eye(2) - A)) * B + D; % Controllable Form
[num, denum] = ss2tf(A, B, C, D); % ss2tf Command Convert A, B, C, D to TF
TF_2 = tf(num, denum);
%% [3] State Transition Matrix
syms t; % Time Domain
PHI_S = inv(s .* eye(2) - A);
PHI_T = ilaplace(PHI_S);
PHI_0 = subs(PHI_T, t, 0); % subs(function, old, new) PHI_0 = I
if (unique(PHI_0 == eye(2)))
  disp('Ø(0) = I');
else
  disp('Ø(0) ≠ I');
end
%% [4] Derivative of State Transition Matrix
PHI_Diff = diff(PHI_T, t);
APHI = A * PHI_T;
if (unique(PHI_Diff == APHI))
  disp('dØ/dt = A.Ø(t)');
else
  disp('dØ/dt ≠ A.Ø(t)');
end
%% [5] Controllability and Observability of this SSR
% Checking Contrillability Δ(Qc) ≠ 0 (Expected Controllable)
Rank_C = rank(ctrb(ss(system)));
degree = length(denum) - 1;
if (Rank_C == degree)
  disp(['System is Controllable with Rank = ', num2str(degree)]);
else
  disp('System is Uncontrollable');
end
```

```
% Checking Observability Δ(Qo) ≠ 0 (Expected Observable)
Rank_O = rank(obsv(ss(system)));
if (Rank_O == degree)
   disp(['System is Observable with Rank = ',num2str(degree)]);
else
   disp('System is Non Observable');
end
%% [6] Unforced (Homogeneous) Solution of the States x(t) and The Unforced Response y(t)
x_0 = [0 ; 1];
x_unforced = PHI_T * x_0;
y_unforced = C * x_unforced;
%% [7] The Unhomogeneous System
U = 1/s;
x_forced = x_unforced + ilaplace((inv(s.* eye(2) - A)) * B * U);
y_forced = C * x_forced;
```

The END Thank You