

ENPM 808Q: Mini Project #1

Due on Monday, November 9, 2015

Dr. Nuno Martins 4:00 PM

Kanishka Ganguly

Contents

Problem 1	3
(A)	3
(B)	5
Problem 2	6
(A)	6
(B)	6
(C)	9
Problem 3	10
(A)	10
(B)	13
Problem 4	17
Q. 6-18	17
Q. 6-19	19
Q. 6-20	20
Q. 6-21	21

Problem 1

Listing 1: Algorithm to Retrieve Parameters.

```

1  %% Initialize Variables %%
2  syms('M','K','rho');
3  %% Plot Samples %%
4  plot(time, samples);
5  %% Solving %%
6  rng('default');
7  r = randi(96);
8  r1 = r+1;
9  r2 = r+2;
10 r3 = r+3;
11 r4 = r+4;
12
13 diff_val1 = (samples(r1)-samples(r))/(time(r1)-time(r)); %Differentiation
    from Slope
14 int_val1 = (0.5*(time(r1)-time(r))*(samples(r1)-samples(r)) +
    ((samples(r))*(time(r1)-time(r)))); %Integration from Area Under Curve
15
16 diff_val2 = (samples(r2)-samples(r1))/(time(r2)-time(r1)); %Differentiation
    from Slope
17 int_val2 = (0.5*(time(r2)-time(r1))*(samples(r2)-samples(r1)) +
    ((samples(r1))*(time(r2)-time(r1)))); %Integration from Area Under Curve
18
19 diff_val3 = (samples(r3)-samples(r2))/(time(r3)-time(r2)); %Differentiation
    from Slope
20 int_val3 = (0.5*(time(r3)-time(r2))*(samples(r3)-samples(r2)) +
    ((samples(r2))*(time(r3)-time(r2)))); %Integration from Area Under Curve
21
22 diff_val4 = (samples(r4)-samples(r3))/(time(r4)-time(r3)); %Differentiation
    from Slope
23 int_val4 = (0.5*(time(r4)-time(r3))*(samples(r4)-samples(r3)) +
    ((samples(r3))*(time(r4)-time(r3)))); %Integration from Area Under Curve
24 %% Forming Equations %%
25 eqn1 = M*(diff_val2-diff_val1) - (time(r1)-time(r)) +
    rho*(samples(r1)-samples(r)) + K*(int_val2-int_val1) == 0;
26 eqn2 = M*(diff_val3-diff_val2) - (time(r2)-time(r1)) +
    rho*(samples(r2)-samples(r1)) + K*(int_val3-int_val2) == 0;
27 eqn3 = M*(diff_val4-diff_val3) - (time(r3)-time(r2)) +
    rho*(samples(r3)-samples(r2)) + K*(int_val4-int_val3) == 0;
28 %% Solving for M, K, rho %%
29 S = solve([eqn1, eqn2, eqn3],[M,K,rho]);
30 M_out = eval(S.M)
31 K_out = eval(S.K)
32 rho_out = eval(S.rho)

```

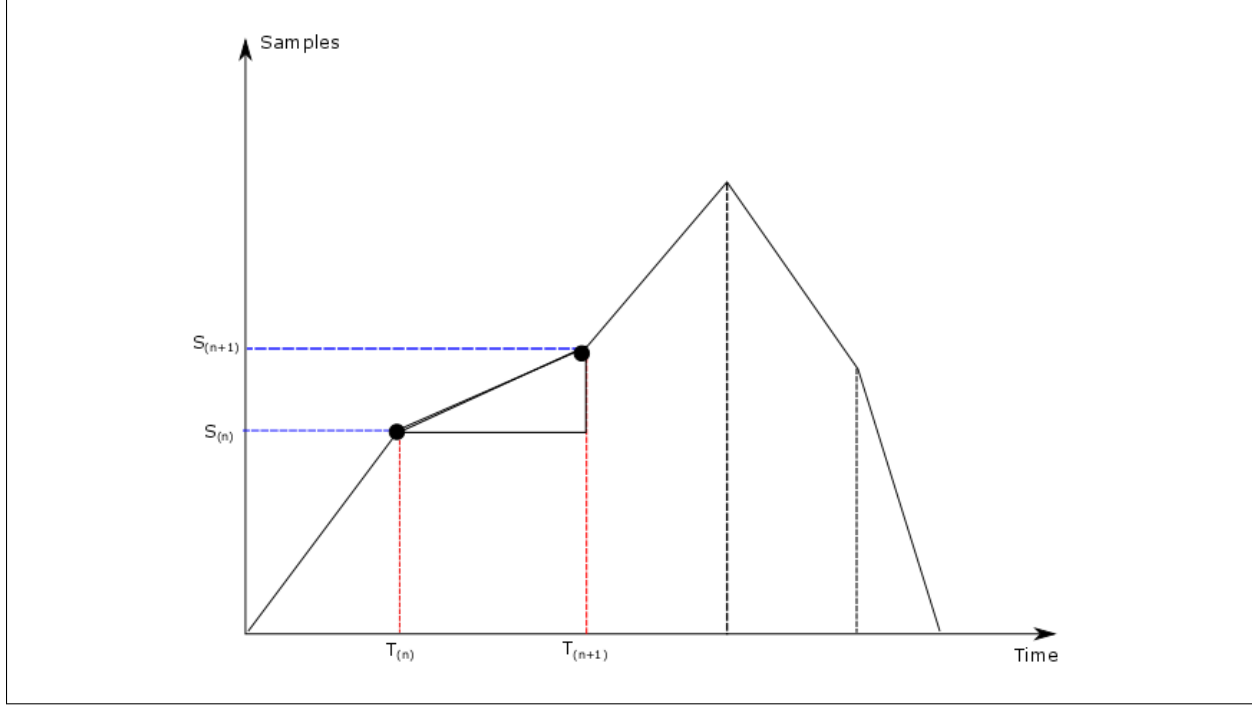
(A)

To determine the constants M, ρ, K , the following algorithm is proposed: For the given equation of the system as:

$$M\ddot{x}(t) + \rho\dot{x}(t) + Kx(t) = F(t), \quad t \geq 0 \quad (1)$$

we have given the time vector \vec{t} and the corresponding sample vector $\vec{x}(\vec{t})$.

Plotting the values of $\vec{x}(\vec{t})$ versus \vec{t} , we get a graph similar to that given below:



Graph of Samples vs. Time

Now, we know that the area under a section of the curve gives us the integral value of the function between the limits chosen.

Also, the slope of the curve between the two selected points gives us the differential value of the function between those two points.

This is shown in the figure marked by the red and blue lines.

The integral value between the two points marked in black is obtained by calculating the sum of the areas of the triangle and the rectangle below the points, as shown. The differential value is calculated by computing the value of $\frac{(y_2 - y_1)}{(x_2 - x_1)}$ where (x_1, y_1) is the first point and (x_2, y_2) is the second point.

Now, given the equation of the system as:

$$M\ddot{x}(t) + \rho\dot{x}(t) + Kx(t) = F(t), \quad t \geq 0 \quad (2)$$

We have

$$M\ddot{x}(t) = F(t) - \rho\dot{x} - Kx \quad (3)$$

From the explanation above, it is clear that we cannot compute the value of $\ddot{x}(t)$. So,

$$M \int_{t_1}^{t_2} \ddot{x}(t) dt = \int_{t_1}^{t_2} F(t) dt - \rho \int_{t_1}^{t_2} \dot{x}(t) dt - K \int_{t_1}^{t_2} x(t) dt \quad (4)$$

$$\implies M\{\dot{x}(t_2) - \dot{x}(t_1)\} = (t_2 - t_1)F - \rho\{x(t_2) - x(t_1)\} - K \int_{t_1}^{t_2} x(t) dt \quad (5)$$

Thus, the equation has been reduced to a set of three unknowns only, i.e. M, K and ρ . The values of $\dot{x}(t)$, $\int x(t)$ and $x(t)$ are calculated graphically, as described above.

Thus, given any four samples, using this algorithm, we obtain a set of three equations having three unknowns.

Simultaneously solving the equations gives us values for the parameters M , K and ρ .

Upon simulation in MATLAB, the values obtained were $M = 0.8988$, $K = 0.9152$ and $\rho = 0.0691$. This comes quite close to the actual values of the parameters, taken as $M = 1$, $K = 1$, $\rho = 0.1$.

Thus, it is possible to determine approximately the values of M , K and ρ from an infinite list of samples.

(B)

As demonstrated above, the algorithm requires at most 4 samples to determine the approximate values of M , K and ρ . However, the higher the sampling rate, i.e. the more number of samples per unit time, the closer the algorithm approaches the true values of M , K and ρ . Given a sufficiently high sampling rate, the error becomes negligible.

Problem 2

(A)

For the system given as:

$$M\ddot{x}(t) + \rho\dot{x}(t) + Kx(t) = F(t), \quad t \geq 0 \quad (6)$$

we have the following state-space equations

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{\rho}{M} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} F \quad (7)$$

This gives us the following state-space parameters:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{\rho}{M} \end{bmatrix} \quad (8)$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (9)$$

$$C = [0 \quad 1] \quad (10)$$

$$D = 0 \quad (11)$$

For controllability, we have the following condition:

Condition 1 A system is controllable if $\text{rank}[B \ AB \ A^2B \ \dots \ A^{n-1}B] = n$

where n is the order of the state.

In this particular case, the system is of order $n = 2$. Thus,

$$\text{rank} [B \ BA] = 2 \quad (12)$$

$$\implies \text{rank} \begin{bmatrix} 0 & 1 \\ 1 & -\frac{\rho}{M} \end{bmatrix} = 2 \quad (13)$$

Now, given this condition, we see that the controllability of the system does not depend on the value of K . Also, irrespective of the value of ρ or M , the rank of the matrix remains the same.

Thus, for any value of M, ρ and K , we have a system that is controllable.

(B)

We are given the parameters as $M = 1, \rho = 0.1, K = 1$. Thus, we have:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{\rho}{M} \end{bmatrix} \quad (14)$$

$$\implies A = \begin{bmatrix} 0 & 1 \\ -1 & -0.1 \end{bmatrix} \quad (15)$$

We have the formula for e^{At} as

$$e^{At} = I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \dots \quad (16)$$

Solving for $A^2, A^3 \dots$ we have

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -0.1 \end{bmatrix} \quad (17)$$

$$A^2 = \begin{bmatrix} -1 & -0.1 \\ 0.1 & -0.99 \end{bmatrix} \quad (18)$$

$$A^3 = \begin{bmatrix} 0.1 & -0.99 \\ 0.99 & 0.199 \end{bmatrix} \quad (19)$$

\vdots

We thus have:

$$e^{At} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -1 & -0.1 \end{bmatrix} t + \begin{bmatrix} -0.5 & -0.05 \\ 0.05 & -0.495 \end{bmatrix} t^2 + \begin{bmatrix} 0.0167 & -0.1650 \\ 0.1650 & 0.0332 \end{bmatrix} t^3 + \dots \quad (20)$$

Now, the formula for calculating the unit step response is given by:

$$y(t) = \int_0^{t_1} C e^{A(t_1-\tau)} B U(\tau) d\tau \quad (21)$$

where B and C are parameters from the state-space given by:

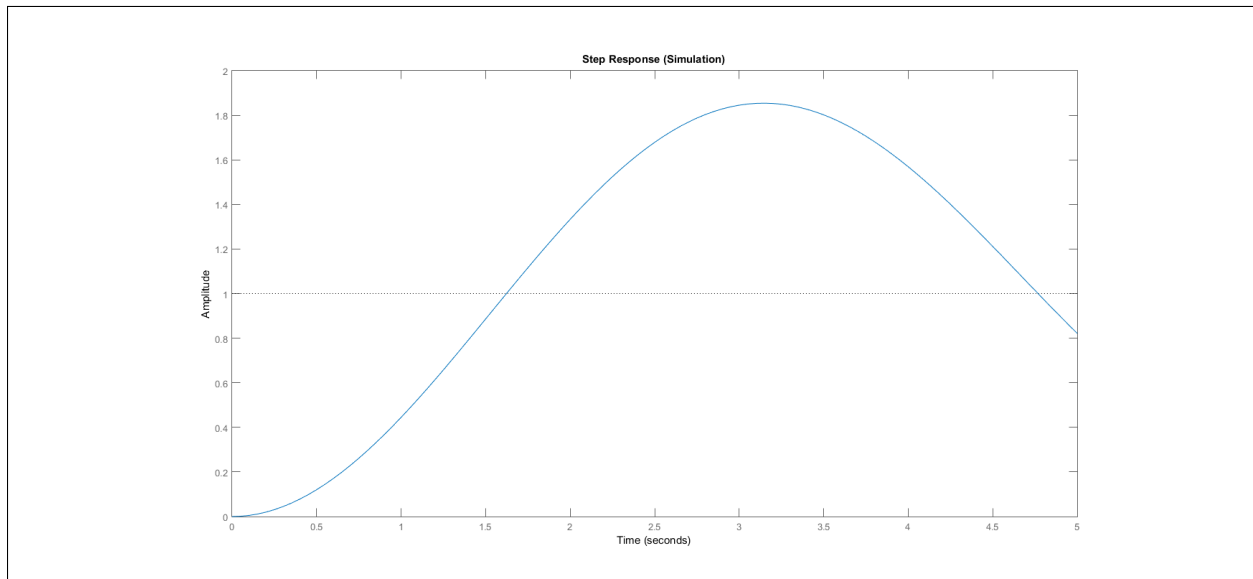
$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (22)$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (23)$$

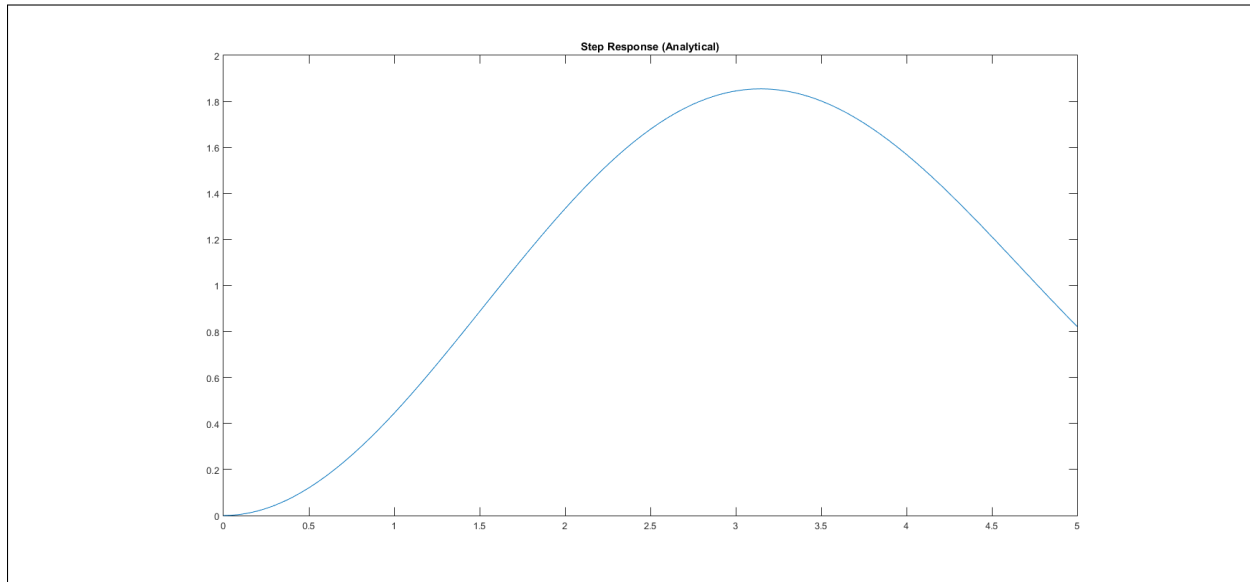
$U(\tau)$ is the unit input, i.e. 1 and t_1 is the upper limit of integration and τ is the time dependent variable.

Proceeding with these values, we expand the series given in Eqn. 16 to 20 terms.

We get results as follows:



Simulated plot of unit step response of system



Analytical plot of unit step response of system

Following is the MATLAB code used to generate the plots:

Listing 2: Unit Step Response Comparison

```

1  %% Initializing State Variable
2  A = [0 1; -1 -0.1];
3  B = [0;1];
4  C = [1 0];
5  D = 0;
6  syms('t');
7  %% Plotting Unit Step Response Through Simulation
8  sys = ss(A,B,C,D);
9  figure
10 step(sys,5);
11 title('Step Response (Simulation)');
12 [y,time] = step(sys,5);
13 %% Creating Lookup Table for Series Calculation
14 eAt_arr = zeros(20,2,2);
15 for i=1:20
16     eAt_arr(i,:,:)= (A^i)/factorial(i);
17 end
18 %% Integrating For Each 'time'
19 vals = zeros();
20 for i=1:size(time)
21     h = time(i);
22     eAt = [1 0; 0 1];
23     for k = 1:20
24         eAt = eAt + permute(eAt_arr(k,:,:).*((h-t)^k),[2,3,1]);
25     end
26     y = C * eAt * B;
27     vals = [vals vpa(int(y,[0 h]),5)];
28 end
29 %% Plotting Unit Step Response Using Analytical Computation
30 figure

```



```

31 plot(time, vals(1:size(time)))
32 title('Step Response (Analytical)');

```

(C)

We have the following equation for obtaining a control law:

$$U(\tau) = B_k^T e^{A^T(t^* - \tau)} W_c(t^*, 0)^{-1} (\vec{X}_d(t^*) - e^{At^*} \vec{X}(0)) \quad (24)$$

where $W_c(t^*, 0)$ is the Controllability Gramian given by:

$$W_c(t^*, 0) = \int_0^{t^*} e^{A\gamma} B_k B_k^T e^{A^T \gamma} d\gamma \quad (25)$$

We are given the following values:

$$\begin{aligned} M &= 1 \\ K &= 1 \\ \rho &= 0.1 \\ \vec{X}(0) &= 1 \\ \vec{X}_d(t^*) &= 0 \\ t^* &= 10 \end{aligned}$$

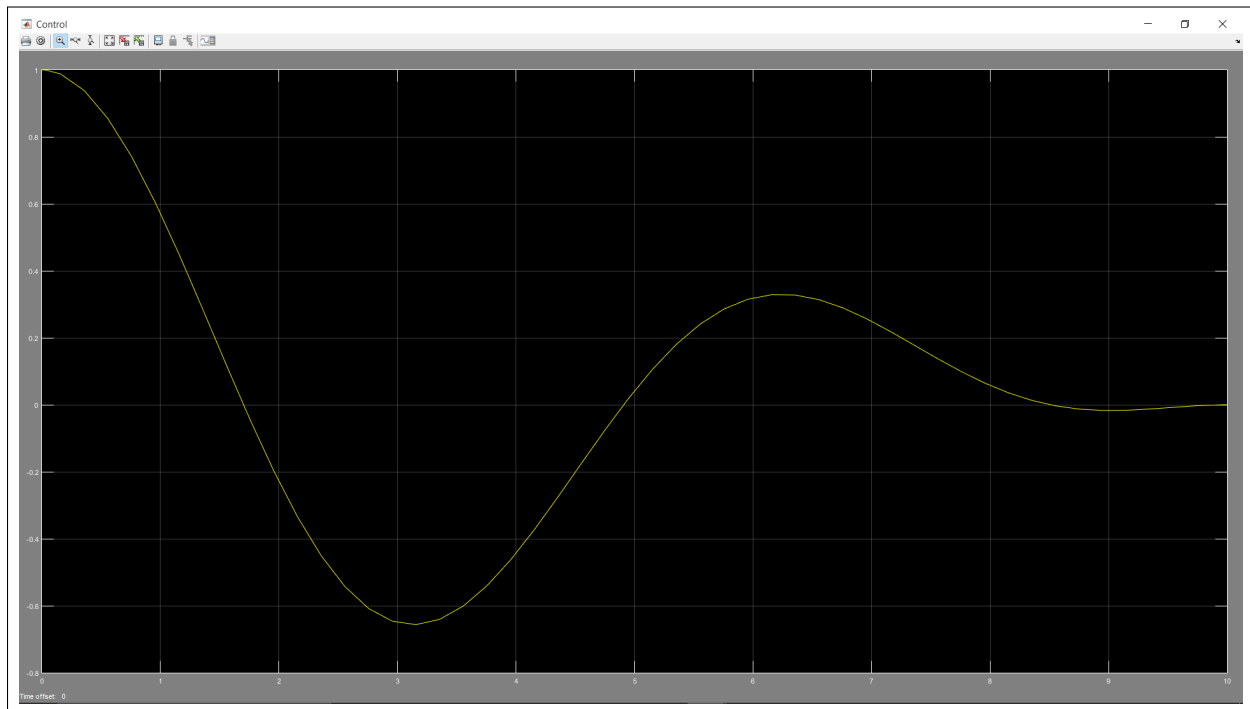
Computing $U(\tau)$ using MATLAB, we obtain:

```

(1.07e+18*cos(399^(1/2)*(tau/20 - 1/2))*exp(tau/20 + 1/2) +
92.41e+15*cos(399^(1/2)*(tau/20 + 1/2))*exp(tau/20 + 1/2) -
1.16e+18*cos(399^(1/2)*(tau/20 - 1/2))*exp(tau/20 + 3/2) +
95.33e+15*399^(1/2)*exp(tau/20 + 1/2)*
sin(399^(1/2)*(tau/20 - 1/2))+2.91e+15*399^(1/2)*
exp(tau/20 + 1/2)*sin(399^(1/2)*(tau/20 + 1/2)) -
98.25e+15*399^(1/2)*exp(tau/20 + 3/2)*sin(399^(1/2)*
(tau/20 - 1/2)))/(45.03e+15*(399*exp(2) - 800*exp(1) +
2*exp(1)*cos(399^(1/2)) + 399)), -(3.69e+18*cos(399^(1/2)*
(tau/20 - 1/2))*exp(tau/20 + 1/2) - 125.6e+15*cos(399^(1/2)*
(tau/20 + 1/2))*exp(tau/20 + 1/2) - 3.57e+18*cos(399^(1/2)*
(tau/20 - 1/2))*exp(tau/20 + 3/2) - 116.72e+15*399^(1/2)*
exp(tau/20 + 1/2)*sin(399^(1/2)*(tau/20 - 1/2)) + 8.94e+15*399^(1/2)*
exp(tau/20 + 1/2)*sin(399^(1/2)*(tau/20 + 1/2)) + 107.77e+15*399^(1/2)*
exp(tau/20 + 3/2)*sin(399^(1/2)*(tau/20 - 1/2)))/
(90.07e+15*(399*exp(2) - 800*
exp(1) + 2*exp(1)*cos(399^(1/2)) + 399))

```

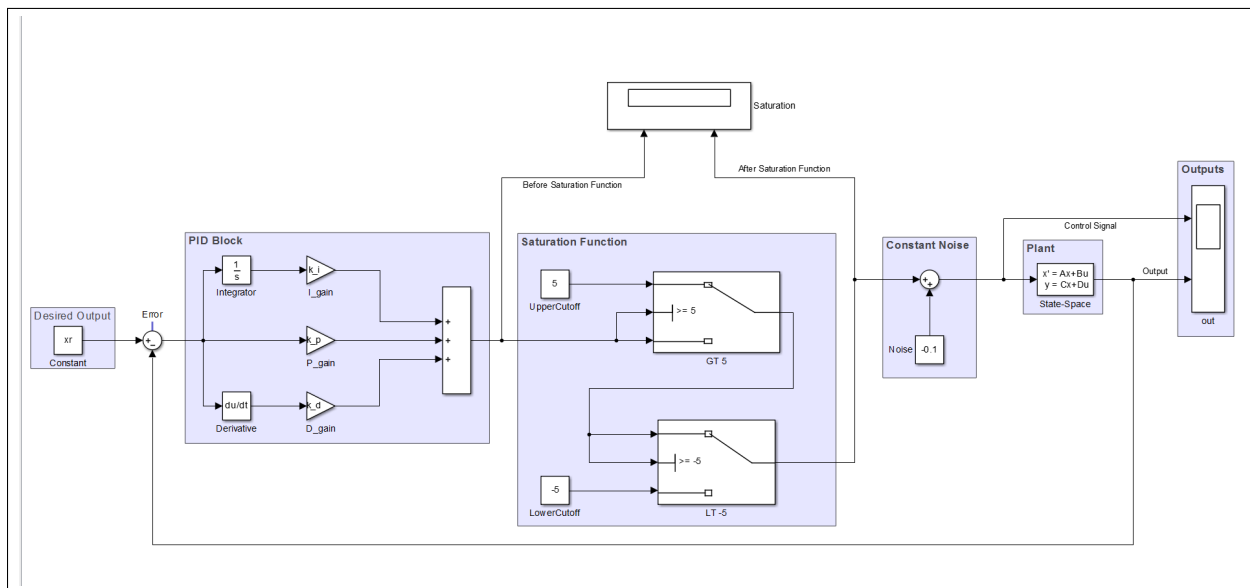
To prove that the control law works as expected, following is the scope output when the above function is given as input to the state-space with a time varying value of τ :

Verifying Control Output $U(t)$

We can see that the state goes from 1 to 0 and converges at time $t \geq 10$, which is the desired output.

Problem 3

(A)



The PID Controller Design

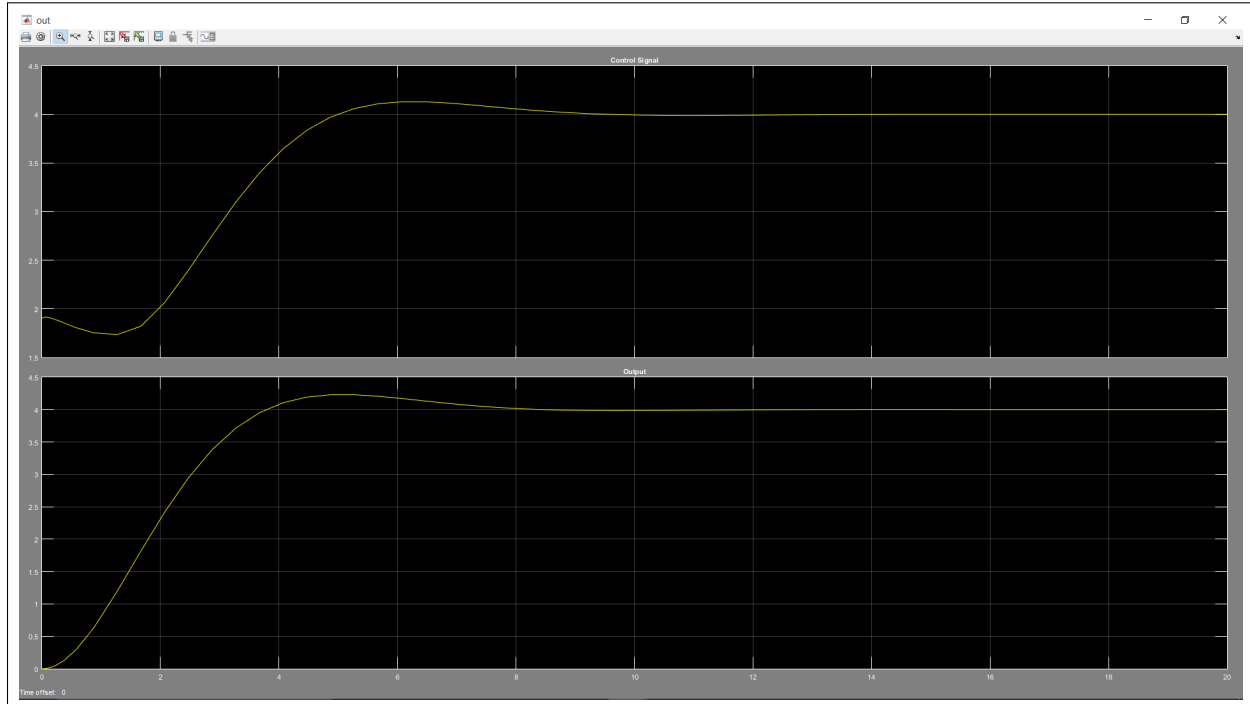
It is given in the problem statement that $M = 1$, $K = 1$ and $\rho = 0.1$. We create our state-space using these

parameters.

Also given is that $x(0) = \dot{x}(0) = 0$, which is set in the state-space block. We also have $-0.1 < F_d < 0.1$, which is a known, constant disturbance as shown in the ‘Constant Noise’ panel of the diagram.

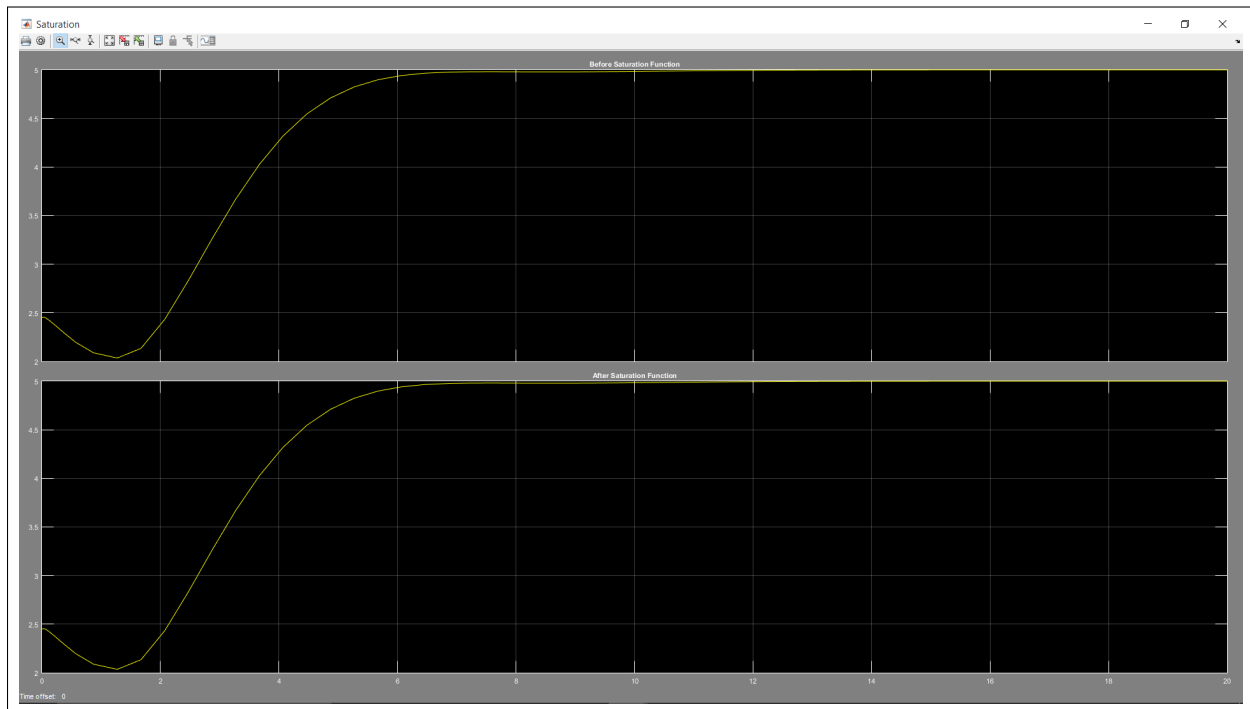
Also shown is the ‘Saturation Function’ block, which is used to satisfy the condition that $F(t)$ saturates at $|F(t)| \geq 5$.

Assuming a reference value $x_r = 4$ and $k_P = 0.5, k_I = 0.6, k_D = 1.6$ as the PID gain values, we get the following output:



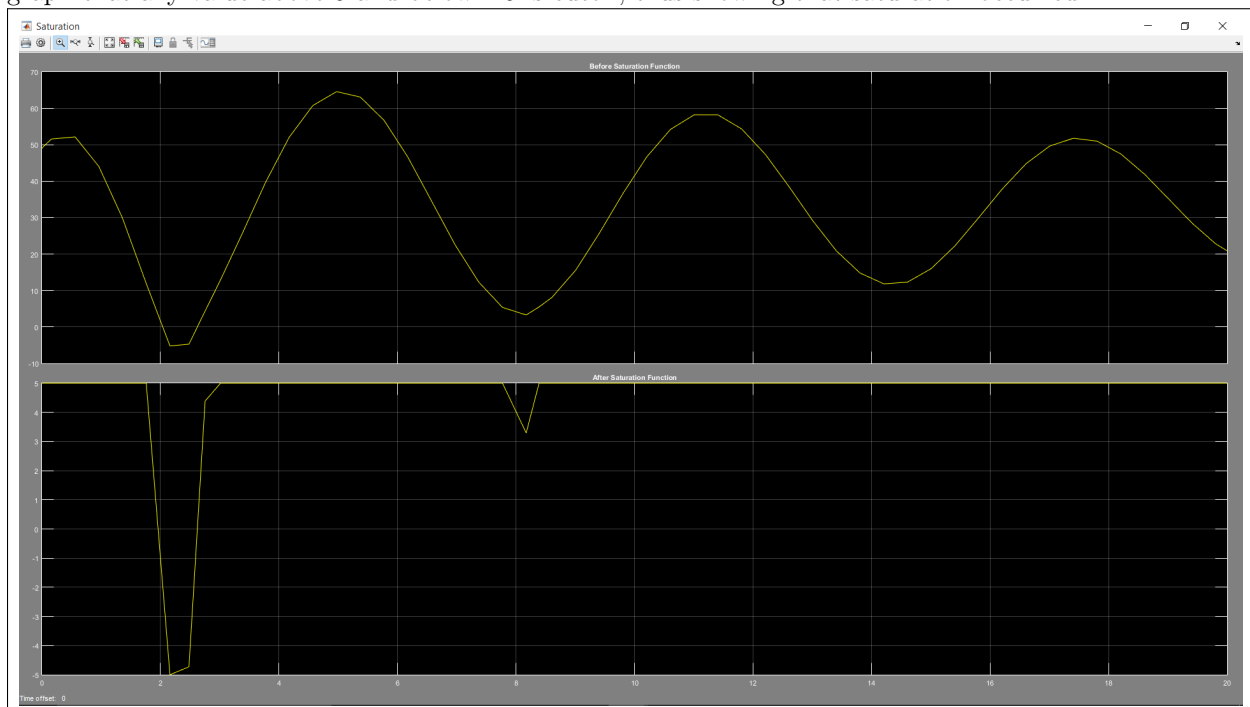
Tracking with zero asymptotic error

For a design for $F(t)$ which never saturates, $k_P = 0.5, k_I = 0.6, k_D = 1.6$ were selected as the PID gain values. A comparison of scope outputs before and after the ‘Saturation Function’ block is shown. They are the same outputs, proving that no saturation/cutoff was done.



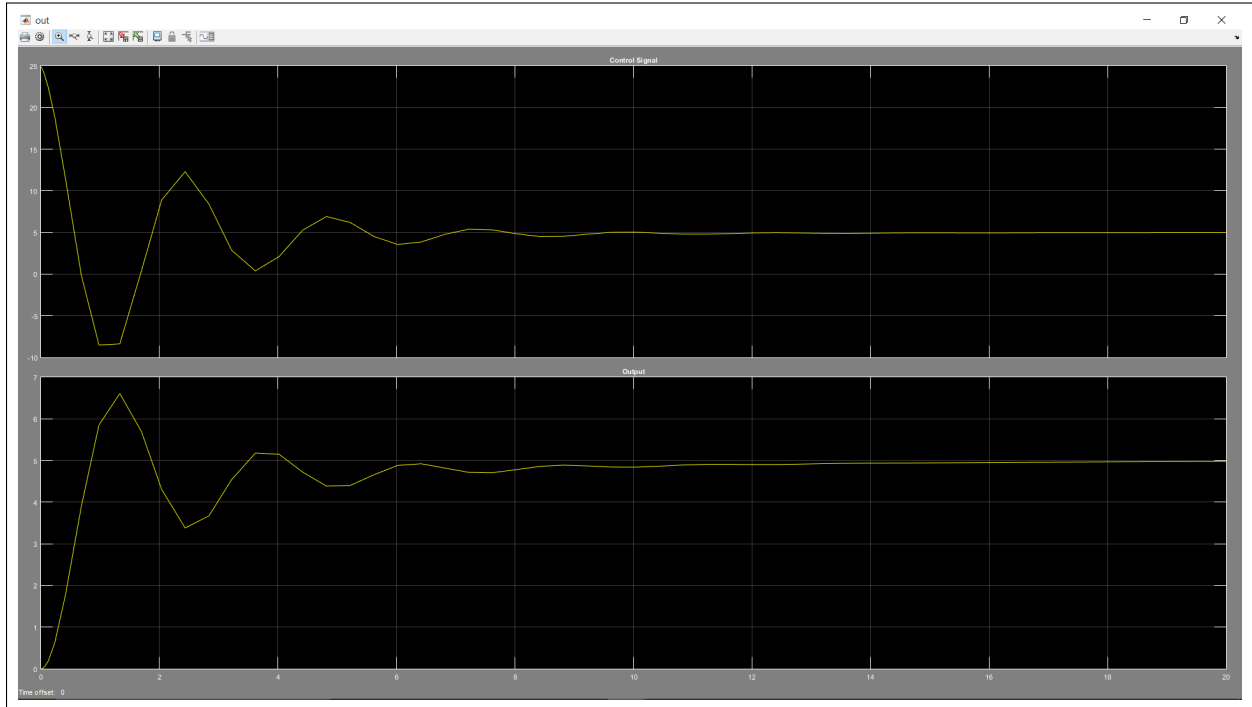
No Saturation of $F(t)$

In order to show saturation, $k_P = 5, k_I = 1, k_D = 1$ were selected as the PID gain values. A similar comparison of scope outputs before and after the 'Saturation Function' block is shown. It is clear from the graph that any value above 5 and below -5 is cutoff, thus showing that saturation occurred.



Saturation of $F(t)$

The output of the system is shown below:

Output with saturation of $F(t)$ **(B)**

Ignoring the saturation condition that $F(t)$ saturates at $|F(t)| \geq 5$, we choose the following state variables for the PID system:

$$X = \begin{bmatrix} x(t) & \dot{x}(t) & \epsilon \end{bmatrix} \quad (26)$$

$$U = x_r(t) \quad (27)$$

where $\epsilon = \int_0^t (x_r - x(t))dt$ is the tracking error and $x_r(t)$ is the desired output.

$$(28)$$

The system is defined by

$$M\ddot{x}(t) + \rho\dot{x}(t) + Kx(t) = F(t) \quad (29)$$

$$\implies M\ddot{x}(t) = F(t) - \rho\dot{x}(t) - Kx(t) \quad (30)$$

We have the general formula for control signal from a PID controller as:

$$F(t) = k_P(x_r - x(t)) - k_D\dot{x}(t) + k_I\epsilon_I - Kx(t) - \rho\dot{x}(t) \quad (31)$$

Substituting Eqn. 31 in Eqn. 30, we get:

$$M\ddot{x}(t) = k_P x_r - (k_P + K)x(t) - (k_D + \rho)\dot{x}(t) + k_I\epsilon_I \quad (32)$$

The general equation for state-space is given by:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}U \quad (33)$$

So, we have

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-(K+k_P)}{M} & \frac{-(k_D+\rho)}{M} & \frac{k_I}{M} \\ -1 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{k_P}{M} \\ 1 \end{bmatrix} U \quad (34)$$

Comparing the above with Eqn. 33, we get

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-(K+k_P)}{M} & \frac{-(k_D+\rho)}{M} & \frac{k_I}{M} \\ -1 & 0 & 0 \end{bmatrix} \quad (35)$$

$$B = \begin{bmatrix} 0 \\ \frac{k_P}{M} \\ 1 \end{bmatrix} \quad (36)$$

This gives us the state-space representation of the closed loop system.

We now check for stability using the following condition:

Condition 2 For a stable system, the eigenvalues of A must have a negative real part.

To calculate the eigenvalues, we have:

$$|\lambda I - A| = 0 \quad (37)$$

$$\Rightarrow \lambda \left[\lambda^2 + \frac{k_D + \rho}{M} \lambda \right] + \lambda \frac{k_P + K}{M} + \frac{k_I}{M} = 0 \quad (38)$$

Considering $k_P = 1.5, k_I = 2, k_D = 1.6$ as the PID gain values, we get

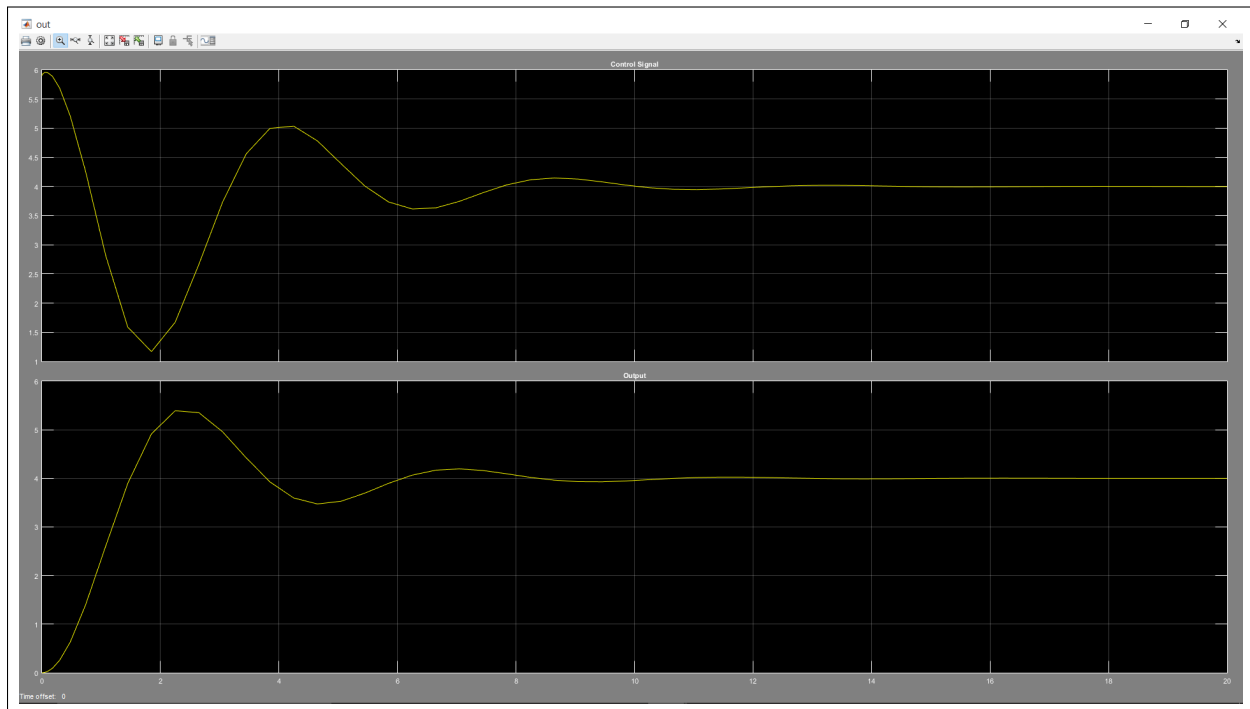
$$\lambda \left[\lambda^2 + \frac{1.6 + 0.1}{1} \lambda \right] + \lambda \frac{1.5 + 1}{1} + \frac{2}{1} = 0 \quad (39)$$

$$\Rightarrow \lambda^3 + 1.7\lambda^2 + 2.5\lambda + 2 = 0 \quad (40)$$

The roots for above polynomial are:

$$\begin{aligned} & -0.3051 + 1.3198i \\ & -0.3051 - 1.3198i \\ & -1.0899 + 0.0000i \end{aligned}$$

It is clear from the eigenvalues that all of them have negative real parts. Thus, we conclude that the system is stable.

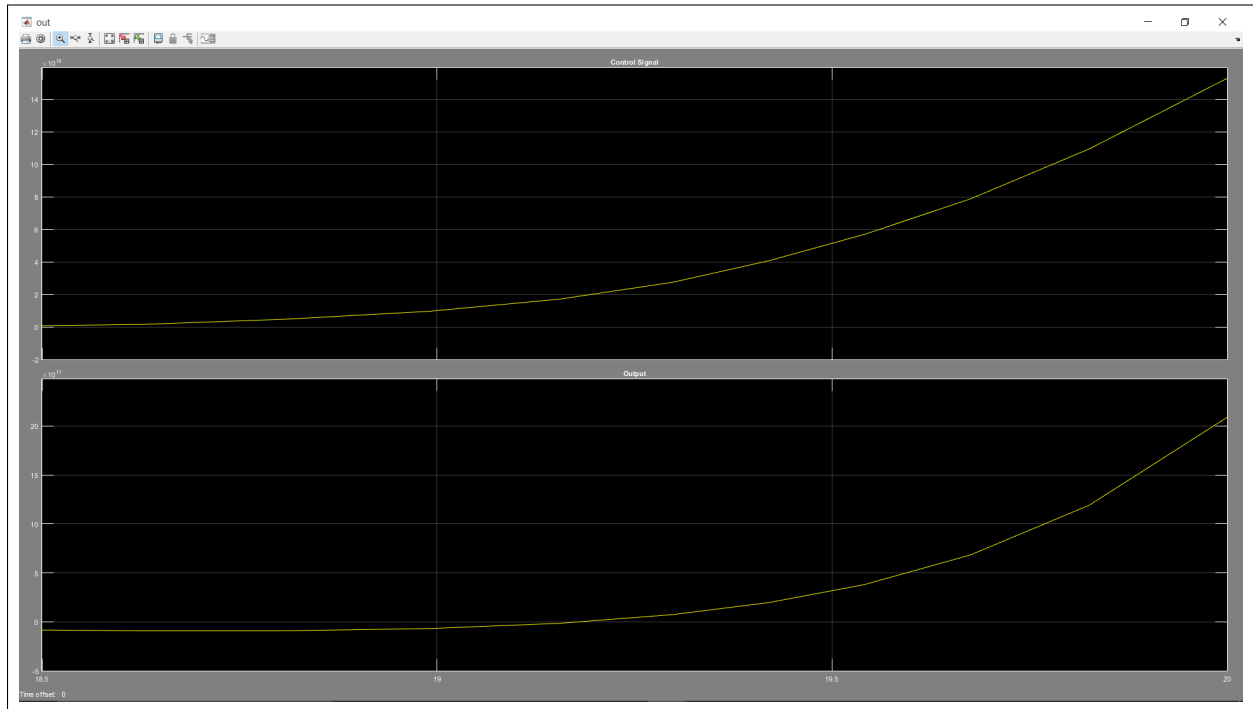


Stable output taking $k_P = 1.5, k_I = 2, k_D = 1.6$ as PID gains

Common intuition might dictate that if a system without PID control is stable, i.e. the A matrix has eigenvalues having negative real parts, it would remain stable even if a PID controller is added, i.e. the A matrix is augmented with the error ϵ . The naive assumption might be that the gain values of the PID controller are independent of the original state-space parameters.

However, the tracking error ϵ is directly affected by the PID gain values, which in turn affects the output of the system since there is a feedback loop. Thus, it is possible to choose values of k_P, k_I and k_D such that the system becomes unstable, even though it was originally a stable system.

Considering a PID controller system without a 'Saturation Function', we take $k_P = 5, k_I = 0, k_D = -5$ as the gain values. The scope output below shows a closed loop system with saturated $F(t)$ giving an unstable output.



Saturated $F(t)$ with unstable output

Problem 4

Q. 6-18

From the problem statement, we are given the following simplified equations of motion:

$$J_1 \ddot{q}_1 = \tau \quad (41)$$

$$J_2 \ddot{q}_2 = \tau \quad (42)$$

For this system of 2^{nd} order equations, we choose the input variables as follows:

$$x_1 = q_1 \quad (43)$$

$$x_2 = \dot{q}_1 \quad (44)$$

$$x_3 = q_2 \quad (45)$$

$$x_4 = \dot{q}_2 \quad (46)$$

$$\Rightarrow X = \begin{bmatrix} q_1 & \dot{q}_1 & q_2 & \dot{q}_2 \end{bmatrix} \quad (47)$$

Also, assuming τ as input, we have the following output variables:

$$q_1 \quad (48)$$

$$q_2 \quad (49)$$

$$\Rightarrow Y = \begin{bmatrix} q_1 & q_2 \end{bmatrix} \quad (50)$$

The general equations for state-space is given as follows:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}U \quad (51)$$

$$\mathbf{Y} = \mathbf{C}\mathbf{X} + \mathbf{D}U \quad (52)$$

So, we have

$$\dot{X} = \left\{ \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \times X \right\} + \left\{ \begin{bmatrix} B_{11} \\ B_{21} \\ B_{31} \\ B_{34} \end{bmatrix} \times U \right\} \quad (53)$$

Thus, we need to select values of $[A_{11} \dots A_{44}]$ and $[B_{11} \dots B_{41}]$ such that Eqn. 54 is satisfied.

So, from Eqn. 51 we have:

$$\dot{X} = \left\{ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} q_1 \\ \dot{q}_1 \\ q_2 \\ \dot{q}_2 \end{bmatrix} \right\} + \left\{ \begin{bmatrix} 0 \\ \frac{1}{J_1} \\ 0 \\ \frac{1}{J_2} \end{bmatrix} \times \tau \right\} \quad (54)$$

Similarly, from Eqn. 52 we get:

$$Y = \left\{ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} q_1 \\ \dot{q}_1 \\ q_2 \\ \dot{q}_2 \end{bmatrix} \right\} + \left\{ 0 \times \tau \right\} \quad (55)$$

Thus, from Eqns. 54 and 55, we get the values of:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (56)$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{J_1} \\ 0 \\ \frac{1}{J_2} \end{bmatrix} \quad (57)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (58)$$

$$D = 0 \quad (59)$$

Now, we consider matrices A and B from Eqns. 56 and 57 to test for controllability of the system. One of the conditions of controllability says that:

Condition 3 *A system is controllable if $\text{rank}[B \ AB \ A^2B \ \dots \ A^{n-1}B] = n$*

where n is the order of the state.

Given this condition and that the order of the system under consideration is $n = 4$, we have:

$$[B \ AB \ A^2B \ A^3B] = \begin{bmatrix} 0 & \frac{1}{J_1} & 0 & 0 \\ \frac{1}{J_1} & 0 & 0 & 0 \\ 0 & \frac{1}{J_2} & 0 & 0 \\ \frac{1}{J_2} & 0 & 0 & 0 \end{bmatrix} \quad (60)$$

It is clear that the rank of the above matrix is 2 which is less than the order of the system $n = 4$.

Thus, the system is uncontrollable.

The obvious implication of the aforementioned result i.e. that the system is uncontrollable, is that the control system will be unable to position the end-effector to the desired position. This will mean that the space shuttle robot arm will be unable to manipulate objects as necessary.

As can be seen from the controllability condition in Condition 3, the rank of the matrix is calculated to be 2 while the order of the system is $n = 4$. This implies that our initial assumption of 4 state variables is incorrect, since a rank of 2 means that only 2 independent variables are needed to represent the system.

Thus, from our initial equations 41 and 42, we can see that only one of them can be used to represent our system. The latter is a dependent equation and can be written in terms of the former.

We have:

$$J_1 \ddot{q}_1 = \tau \quad (61)$$

$$\implies \ddot{q}_1 = \frac{\tau}{J_1} \quad (62)$$

$$\implies q_1 = \iint \frac{\tau}{J_1} \quad (63)$$

Similarly, we have:

$$J_2 \ddot{q}_2 = \tau \quad (64)$$

$$\implies \ddot{q}_2 = \frac{\tau}{J_2} \quad (65)$$

Substituting from 61, we get

$$\ddot{q}_2 = \frac{\ddot{q}_1 J_1}{J_2} \quad (66)$$

From 63

$$(67)$$

$$\Rightarrow \ddot{q}_2 = \iint \frac{\tau}{J_1} \times \frac{J_1}{J_2} \quad (68)$$

$$\Rightarrow q_2 = \iint \left\{ \left[\iint \frac{\tau}{J_1} \right] \times \left[\frac{J_1}{J_2} \right] \right\} \quad (69)$$

Thus, we see that the value of q_2 is dependent on the force τ and the ratio of the inertias J_1 and J_2 . This means that our initial assumption of state variables can be reduced to 2 variables q_1 and \dot{q}_1 . This would result in the rank of the matrix $[B \ A^2 B] = 2$, thus making the system controllable.

Q. 6-19

We have, from problem statement:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \end{bmatrix} u \quad (70)$$

where

$$u = k_1 x_1 + k_2 x_2 \quad (71)$$

We need to compute a linear state feedback control such that the closed loop system has poles at $s = -2, 2$. Thus, we need to obtain values for k_1 and k_2 .

We can write u as

$$u = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (72)$$

Substituting Eqn. 72, we get:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (73)$$

$$\Rightarrow A + Bk = \left\{ \begin{bmatrix} 1 & -3 \\ 1 & -2 \end{bmatrix} + \begin{bmatrix} k_1 & k_2 \\ -2k_1 & -2k_2 \end{bmatrix} \right\} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (74)$$

$$\Rightarrow A + Bk = \begin{bmatrix} k_1 + 1 & k_2 - 3 \\ 1 - 2k_1 & -2 - 2k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (75)$$

The eigenvalues (poles) are given by

$$\det(\lambda I - A + Bk) = 0 \quad (76)$$

$$\Rightarrow \lambda^2 + k_1(-8 - \lambda) + k_2(2\lambda - 3) + \lambda + 1 = 0 \quad (77)$$

Given that the eigenvalues are 2, -2

$$k_1 = \frac{13}{19} \quad (78)$$

$$k_2 = \frac{-3}{19} \quad (79)$$

$\therefore u = \frac{13}{19}x_1 + \frac{-3}{19}x_2$ is the required linear state feedback control.

Q. 6-20

We have, from problem statement:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (80)$$

where

$$u = k_1 x_1 + k_2 x_2 \quad (81)$$

We need to compute a linear state feedback control such that the closed loop system has poles at $s = -2, 2$. Thus, we need to obtain values for k_1 and k_2 .

We can write u as

$$u = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (82)$$

Substituting Eqn. 82, we get:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (83)$$

$$\Rightarrow A + Bk = \left\{ \begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ k_1 & k_2 \end{bmatrix} \right\} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (84)$$

$$\Rightarrow A + Bk = \begin{bmatrix} -1 & 0 \\ k_1 & k_2 + 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (85)$$

The eigenvalues (poles) are given by

$$\det(\lambda I - A + Bk) = 0 \quad (86)$$

$$\Rightarrow -(\lambda + 1)(k_2 - \lambda + 2) = 0 \quad (87)$$

Considering given eigenvalue 2

$$k_1 \in \mathbb{Z} \quad (88)$$

$$k_2 = 0 \quad (89)$$

Considering given eigenvalue -2

$$k_1 \in \mathbb{Z} \quad (90)$$

$$k_2 = -4 \quad (91)$$

However, from Eqn. 87, it is clear that one of the eigenvalues is defined at $\lambda = -1$.

This implies that we cannot place both closed loop poles at -2.

Thus, we now have two sets of eigenvalues for the system as follows:

$$\lambda = -1, 2 \quad (92)$$

$$\lambda = -1, -2 \quad (93)$$

The conditions for stability can be analyzed from the values of the eigenvalues as:

Condition 4 For a stable system, the eigenvalues need to be negative and as far from the origin as possible.

Then, it is evident that the choice of poles in Eqn. 92 does not satisfy the stability condition since one of the poles is +2.

On the other hand, the choice of poles in Eqn. 93 makes the system stable since both values are negative and far from the origin.

Thus, the system is stable for poles placed at -1, -2 but not at poles placed at -1, 2.

Q. 6-21

We have, from problem statement:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (94)$$

where

$$u = k_1 x_1 + k_2 x_2 \quad (95)$$

We need to compute a linear state feedback control such that the closed loop system has poles at $s = -2, 2$. Thus, we need to obtain values for k_1 and k_2 .

We can write u as

$$u = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (96)$$

Substituting Eqn. 96, we get:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (97)$$

$$\implies A + Bk = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ k_1 & k_2 \end{bmatrix} \right\} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (98)$$

$$\implies A + Bk = \begin{bmatrix} 1 & 0 \\ k_1 & k_2 + 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (99)$$

The eigenvalues (poles) are given by

$$\det(\lambda I - A + Bk) = 0 \quad (100)$$

$$\implies -(\lambda - 1)(k_2 - \lambda + 2) = 0 \quad (101)$$

Considering given eigenvalue 2

$$k_1 \in \mathbb{Z} \quad (102)$$

$$k_2 = 0 \quad (103)$$

Considering given eigenvalue -2

$$k_1 \in \mathbb{Z} \quad (104)$$

$$k_2 = -4 \quad (105)$$

However, from Eqn. 101, it is clear that one of the eigenvalues is defined at $\lambda = 1$.

This implies that we cannot place both closed loop poles at -2.

Thus, we now have two sets of eigenvalues for the system as follows:

$$\lambda = +1, 2 \quad (106)$$

$$\lambda = +1, -2 \quad (107)$$

The conditions for stability can be analyzed from the values of the eigenvalues as:

Condition 5 For a stable system, the eigenvalues need to be negative and as far from the origin as possible.

Then, it is evident that the choice of poles in Eqn. 106 does not satisfy the stability condition since both of the poles are positive at +1 and +2.

Similarly, the choice of poles in Eqn. 107 also makes the system unstable since one of the values is positive at +1.

Thus, the system is unstable for poles placed at +1, +2 and also for poles placed at +1, -2.