دانشگاه
خواجه نصیرالدین طوسی
K. N. Toosi University
of Technology

MPC
4^{th} Assignment

Reza Shahriari, 40308094

January 2, 2025

1 Problem 1

1.1 Part A

For designing the MPC Controller we first have to define the state space representation of the matrix in MATLAB. In the following code, the system is defined and discretized.

```
function [xplus,yd]= SystemDynamics(xd,ud)

% Parameter Definition
M_S = 2500;
M_U = 320;
k_S = 80000;
k_U = 500000;
c_s = 350;
c_U = 15020;

A = [0      1         0            0;
     0 -c_s/M_S -k_S/M_S  c_s/M_S;
     0      1           0        -1;
     -k_U/M_U c_s/M_U (k_S+k_U)/M_U -(c_s+c_U)/M_U];


B = [0 ; 1/M_S ; 0 ;1/M_U];

C = [1 0 0 0; 0 1 0 0; 0 0 1 0];

Ad = [0.9241    0.0945   -0.0672    0.0013;
   -2.0392    0.8450   -0.6570    0.0178;
    0.8830    0.0843   -0.1675    0.0021;
    1.2927    0.1390   -1.8258   -0.0591];

Bd = [    0.1053;
    2.2534;
   -0.8123;
  -51.0695];

Cd = C;

xplus = Ad*xd + Bd*ud;
yd = Cd*xd;
```

Then the system is defined in a Matlab Function, in SIMULINK. The SIMULINK configuration is as follows:

The disturbance is described as follows:

```
function y = Disturbance(t)

```

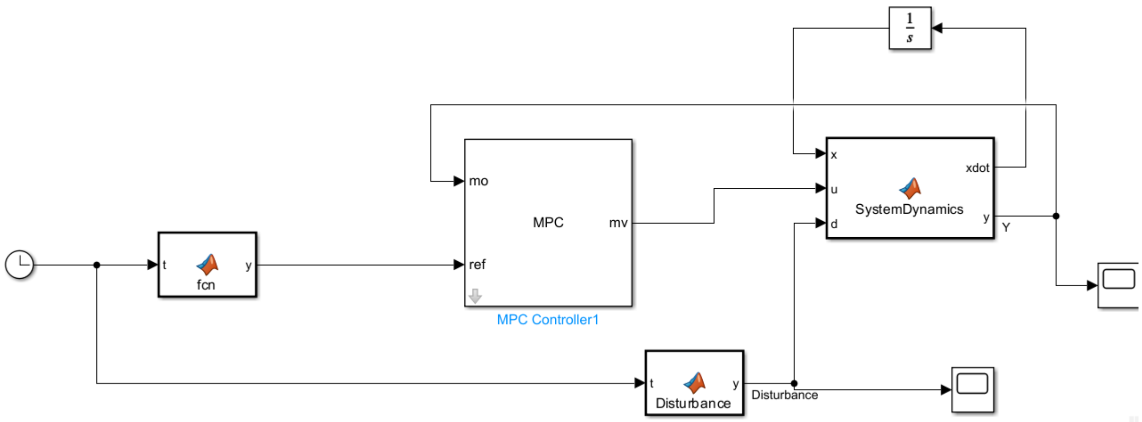
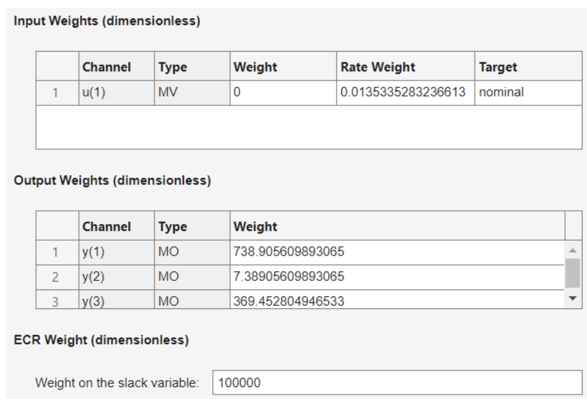


Figure 1: The MPC Configuration



Input Weights (dimensionless)

	Channel	Type	Weight	Rate Weight	Target
1	u(1)	MV	0	0.0135335283236613	nominal

Output Weights (dimensionless)

	Channel	Type	Weight
1	y(1)	MO	738.905609893065
2	y(2)	MO	7.38905609893065
3	y(3)	MO	369.452804946533

ECR Weight (dimensionless)

Weight on the slack variable: 100000

Figure 2: Shows the weights

```
if t>=4 & t<5
    y = t-4;
elseif t>=5 & t<7
    y = 1;
elseif t>=7 & t<8
    y = -t+8;
else
    y = 0;
end
```

The Weights are defined as follows:
The following shows the results of Linear MPC utilized on the system.
The root mean square error of x_1 and x_3 can be calculated using following configuration.
The Output of PID controller is show in figure 8 to compare with the results of MPC

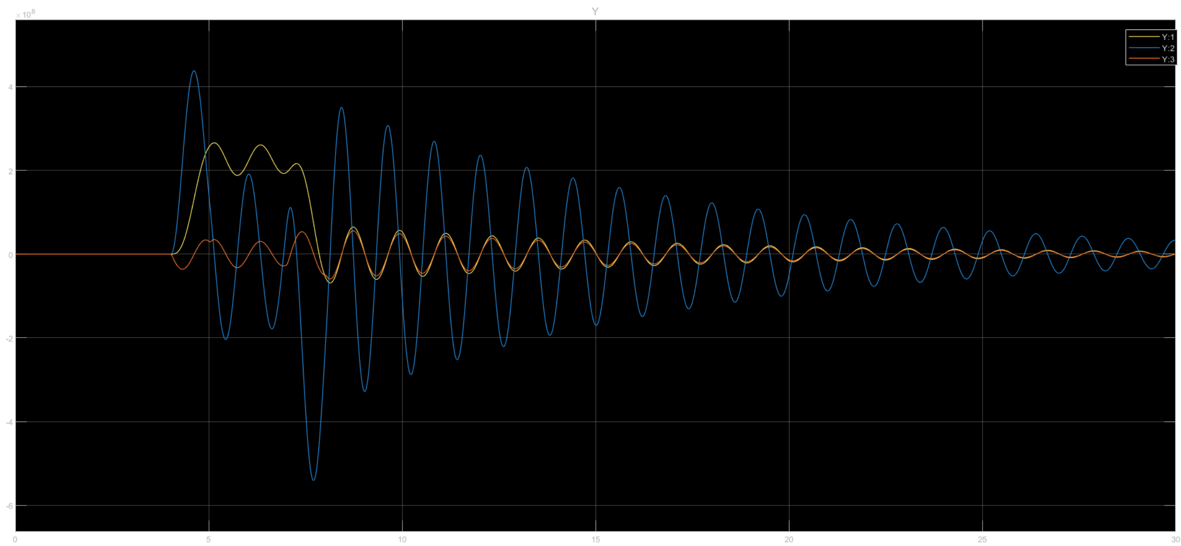


Figure 3: Shows the results of MPC

The PID Parameters are as shown in figure 7:

The Control effort of PID is shown in figure 9 to compare with results of MPC

The RMSE of X1 and X3 is given in figures 10 and 11

The control effort of PID shows a better signal than the Linear MPC and the output shows a better disturbance rejection, and shows a better RMSE in X1 and X3.

1.2 Part B

The tube MPC is simulated in this section, with adding a PID to the Linear MPC we previously had 12, The Configuration bellow is utilized, It is worth noting that the previous linear MPC parameters are consistent with what we previously had, but the PID is tuned again the configuration in figure 13 is presented as the new PID parameters.

The system output is given in 14. It shows magnificent disturbance rejection in the first and second output, but for the third output, not only was the disturbance not rejected, but it was amplified in the opposite direction. These results seem odd.

In figure 15, the control effort is shown.

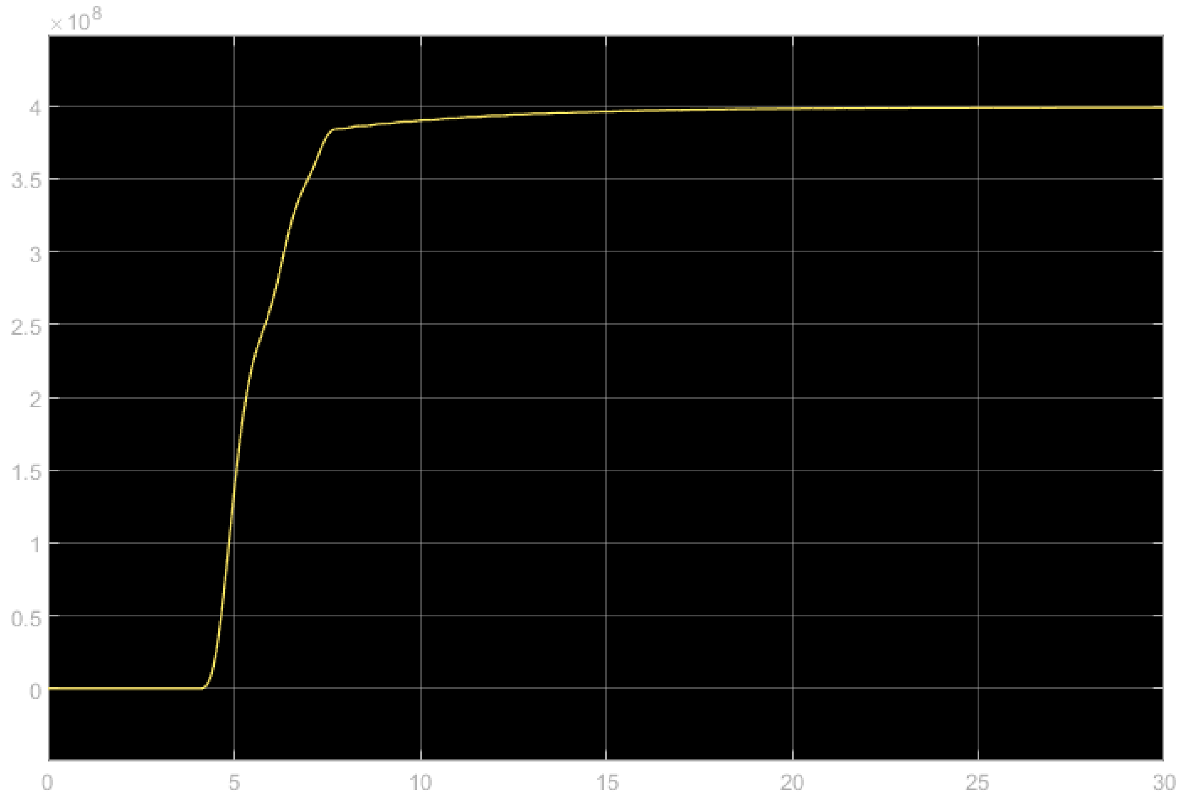


Figure 4: Root Mean Square Error of X1

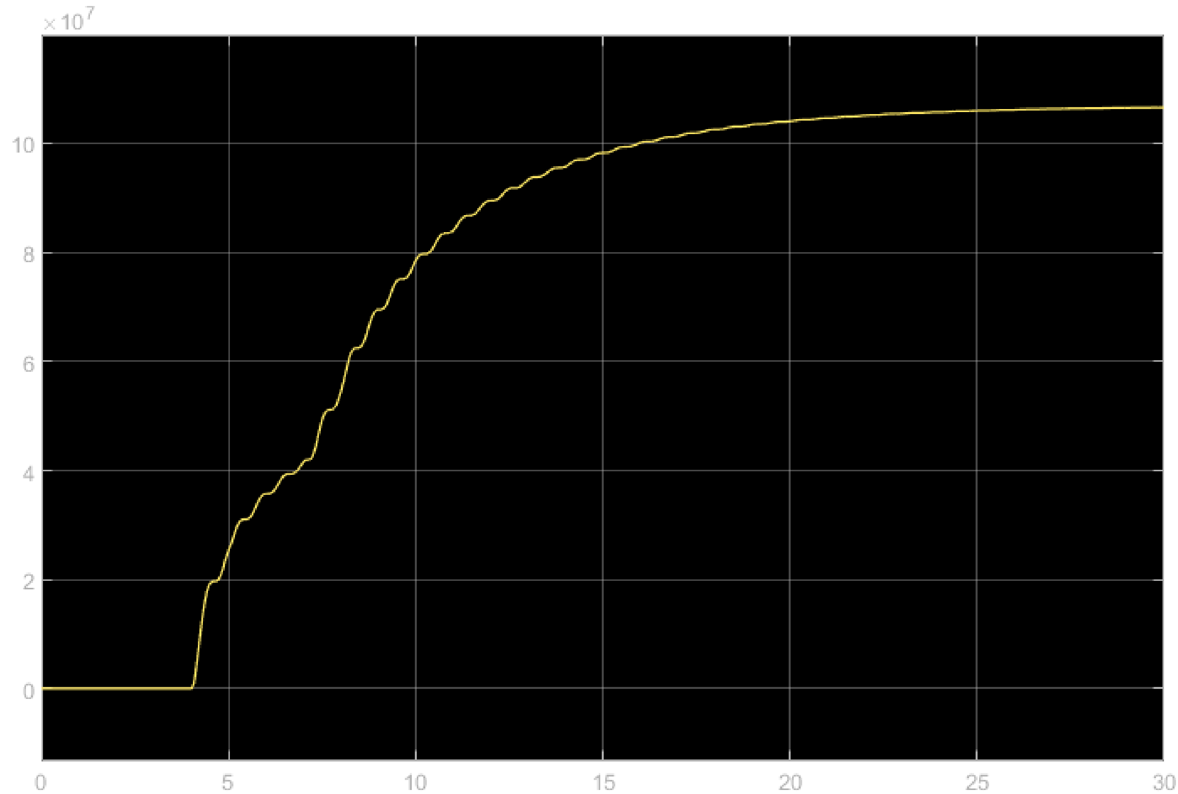


Figure 5: Root Mean Square Error of X2

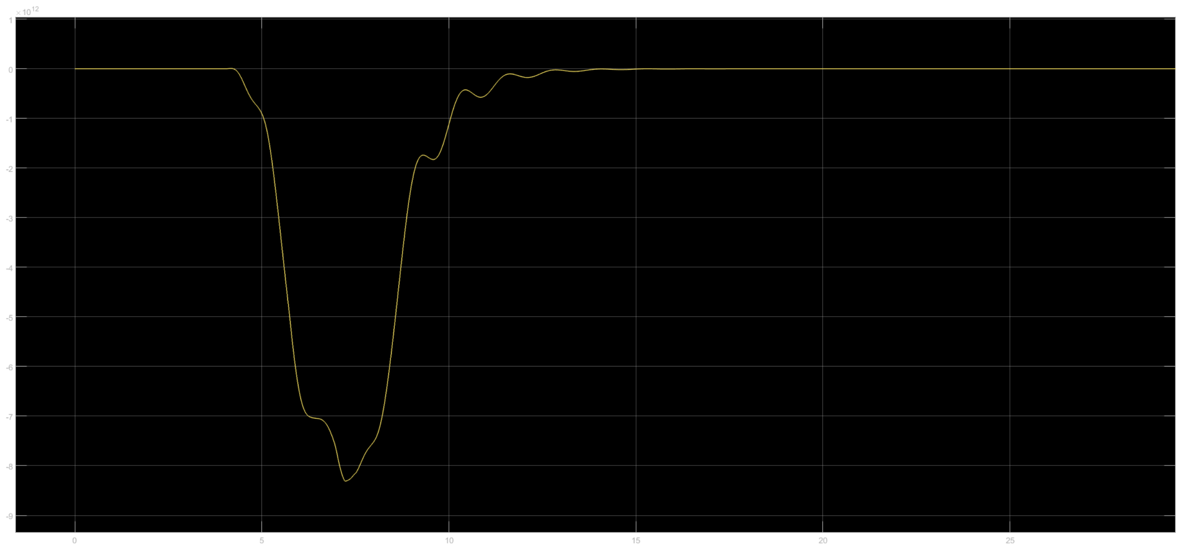


Figure 6: The Control Effort of the system

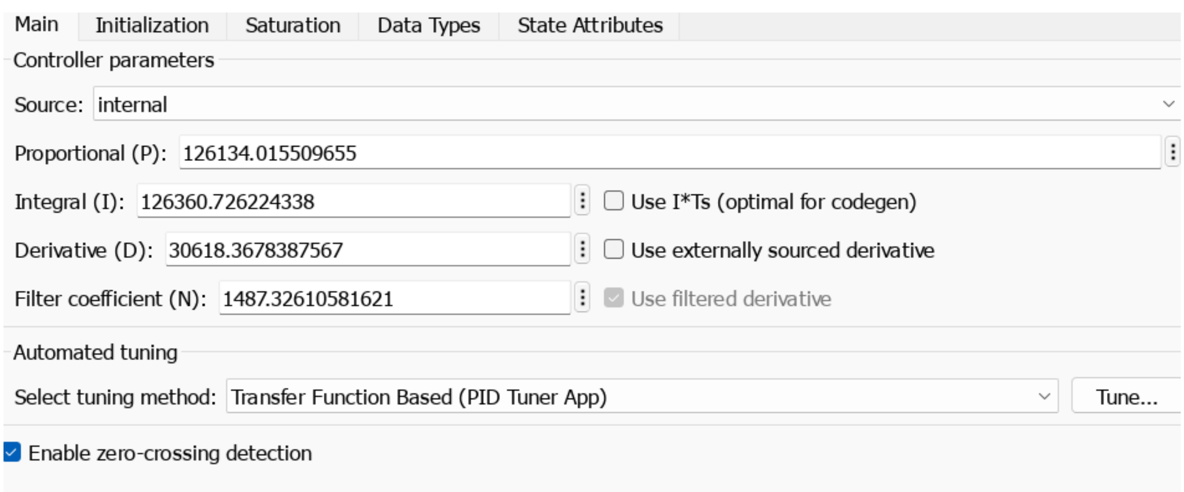


Figure 7: The PID Parameters

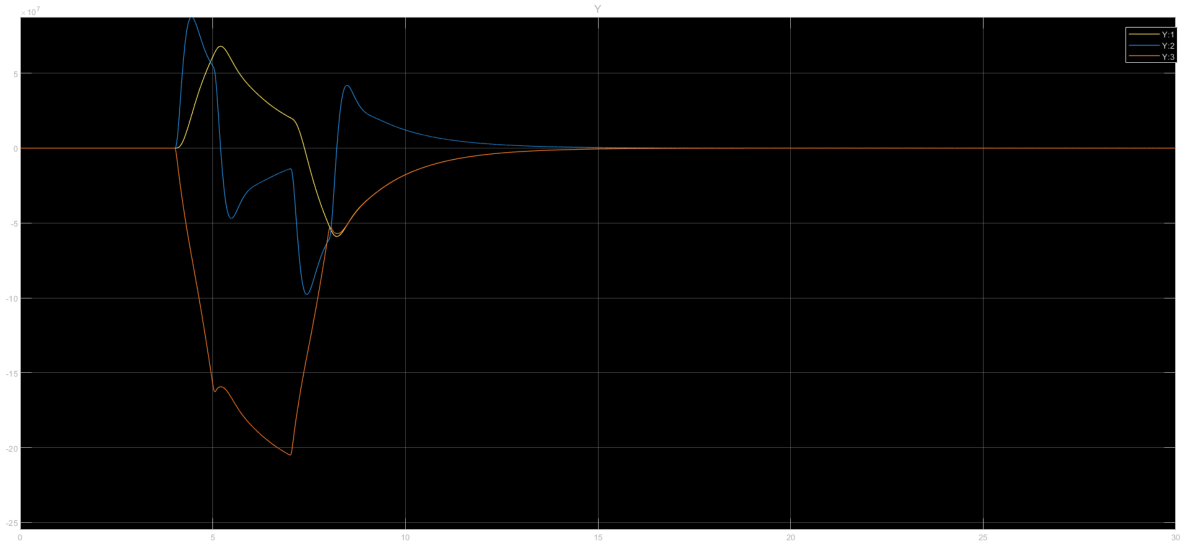


Figure 8: PID Controller output

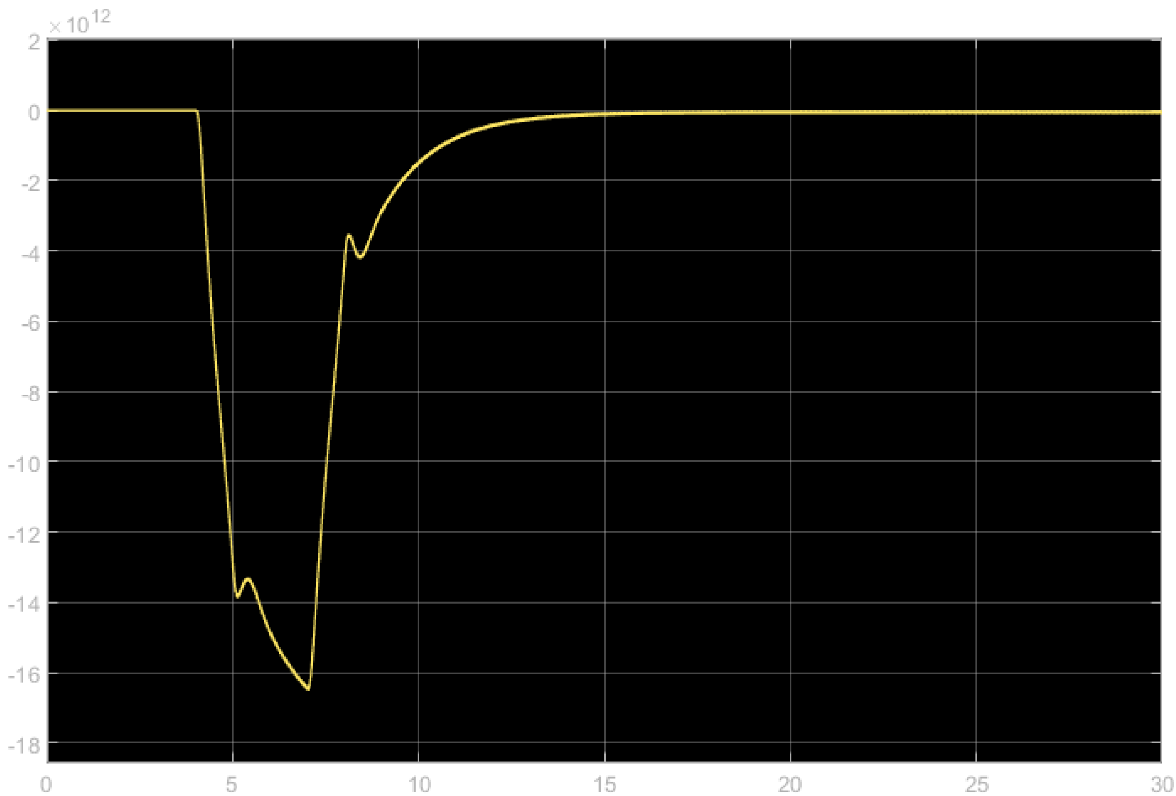


Figure 9: The Control Effort of PID

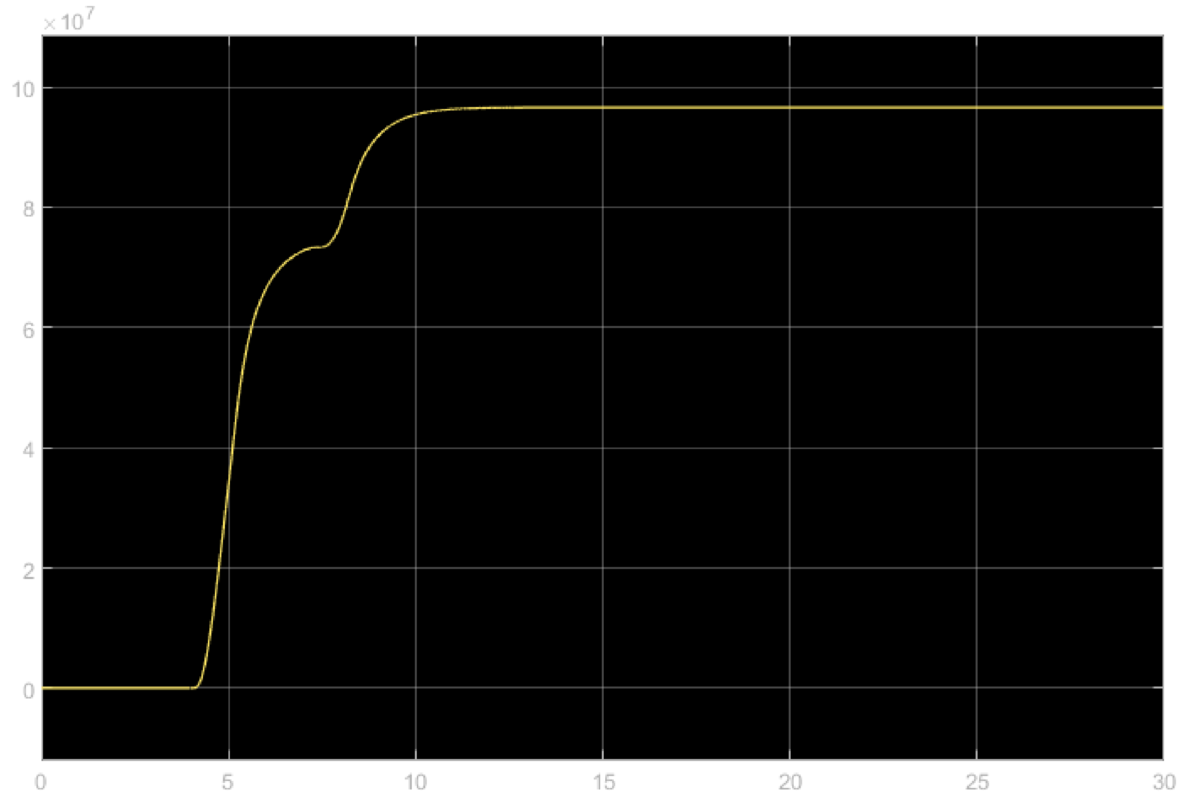


Figure 10: RMSE of X1

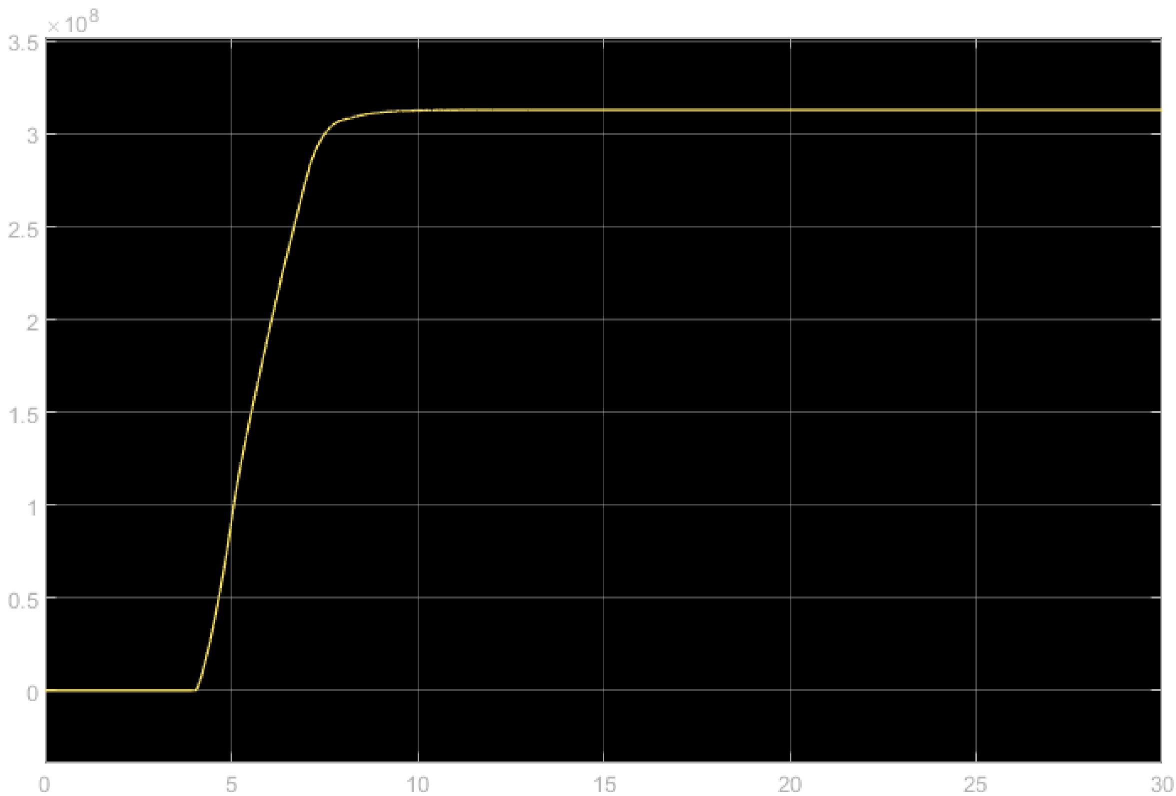


Figure 11: RMSE of X3

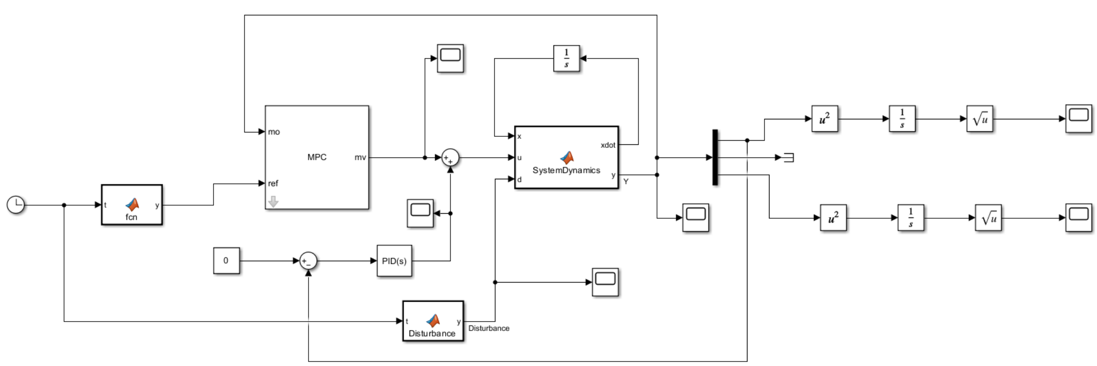


Figure 12: The Configuration of Tube MPC

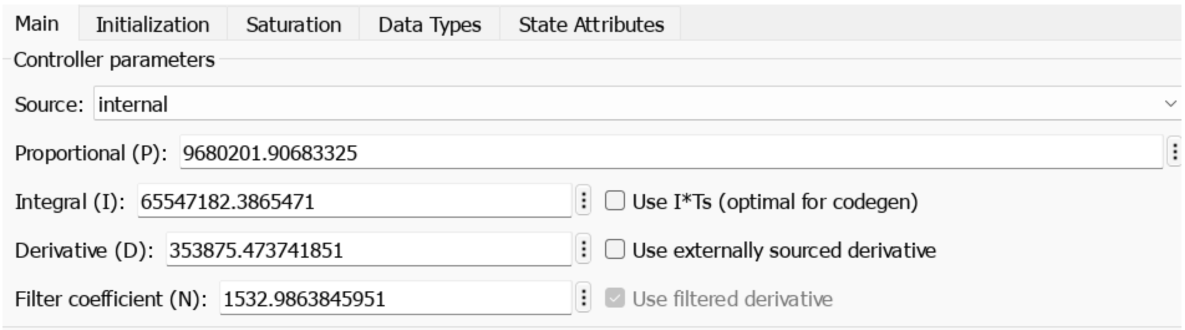


Figure 13: The new PID tuned

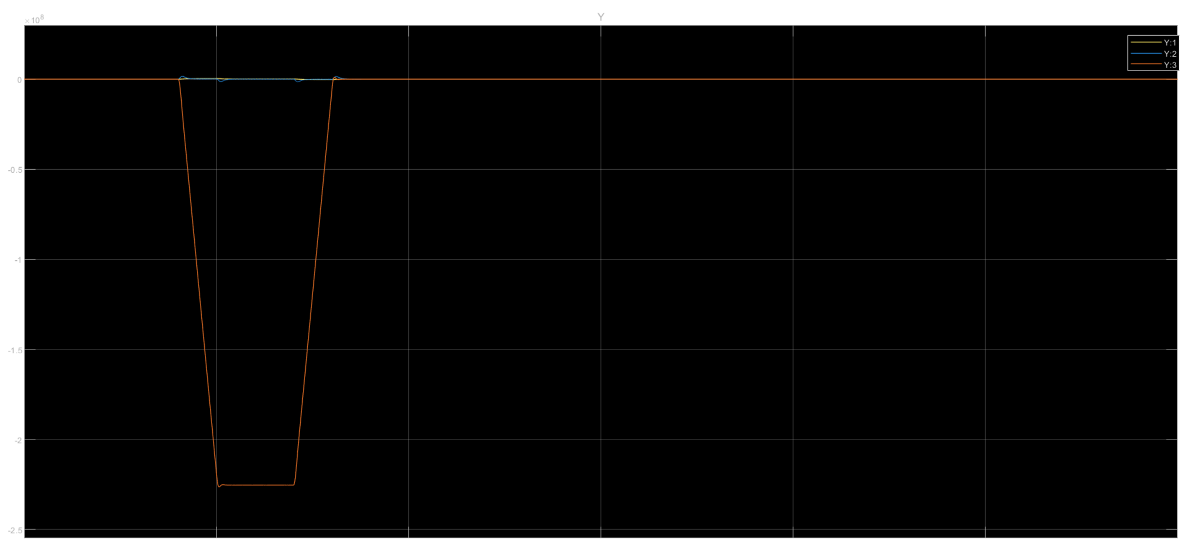


Figure 14: The results of Tube MPC

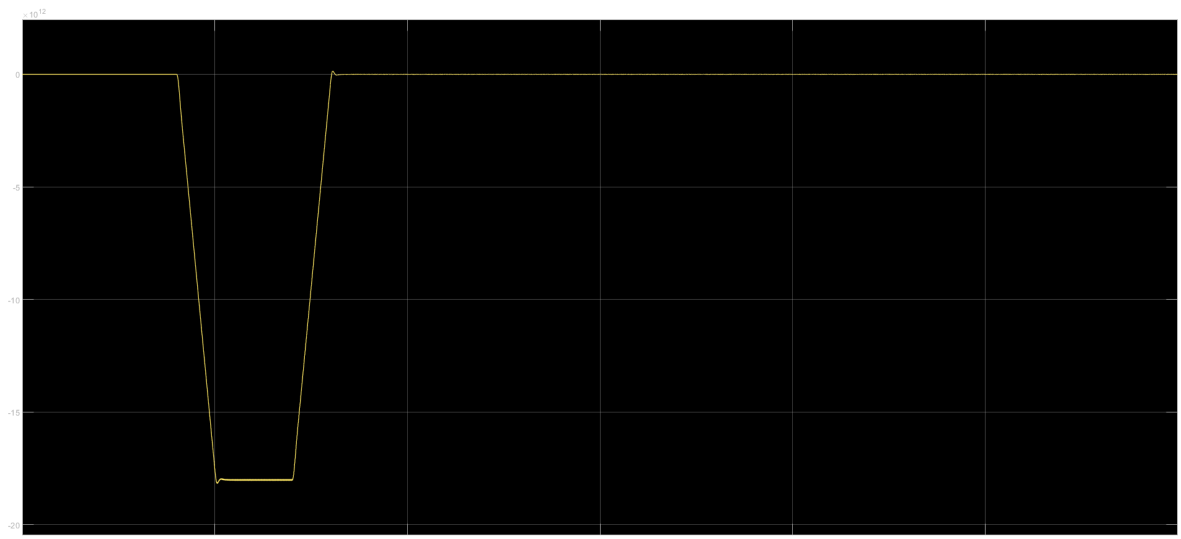


Figure 15: PID+MPC Control Effort

1.3 Part C

For Tuning the explicit MPC, what we should first design the Linear MPC. This we have designed in Part A. Now in part C, to make the Linear MPC explicit we run the following two commands:

```
range = generateExplicitRange(mpc1)
explicit = generateExplicitMPC(mpc1,range)
```

This will create an explicit MPC object named "explicit". The range structure has seven states, in which the range of explicit making is defined. We have defined the min of each state to zero and its max to 1.

```
Explicit MPC Controller
---------------------------------------------
Controller sample time:    0.01 (seconds)
Polyhedral regions:        70
Number of parameters:      7
Is solution simplified:    No
State Estimation:          Default Kalman gain
---------------------------------------------
Type 'explicit.MPC' for the original implicit MPC design.
Type 'explicit.Range' for the valid range of parameters.
Type 'explicit.OptimizationOptions' for the options used in multi-parametric QP computation
Type 'explicit.PiecewiseAffineSolution' for regions and gain in each solution.
```

Figure 16: The results of Explicit MPC

In figure 16 the results of the explicit MPC such as regions, type of filtering use, etc. are presented. The explicit MPC configuration follows like what we had with Linear MPC. The new configuration is shown in 17

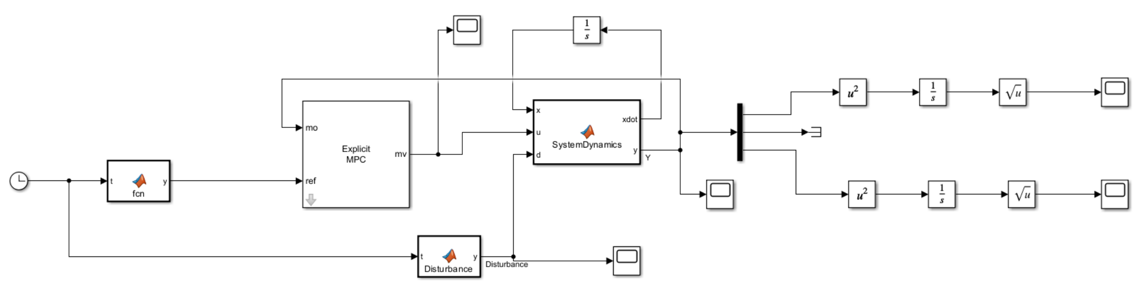


Figure 17: The Configuration of Explicit MPC

The result is also quite similar with what we had with Linear MPC in Part A. In figure 18, the results of explicit MPC's output is presented, in 19, the control effort are shown. and figure 20 a) shows the RMSE of X1 and b) shows the RMSE of X3 for when we have used the explicit MPC.

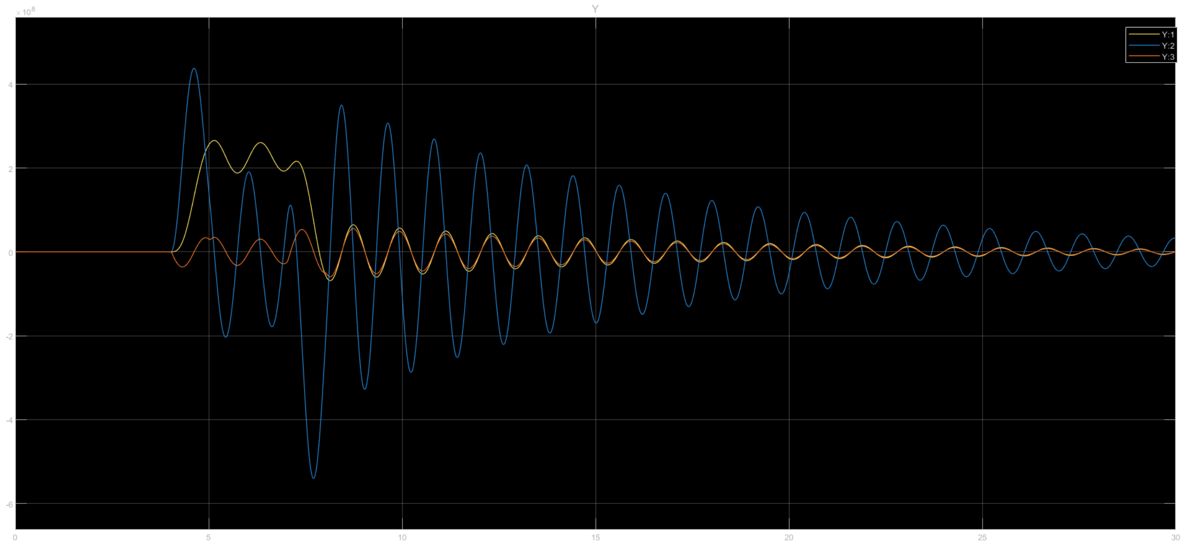


Figure 18: The output of Explicit MPC

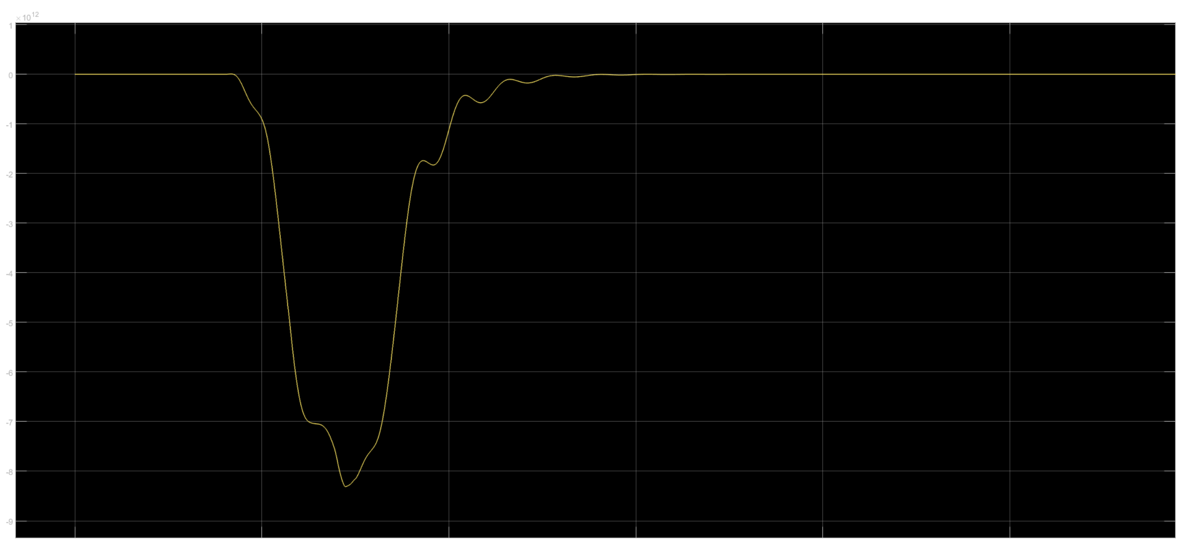


Figure 19: The control effort of Explicit MPC

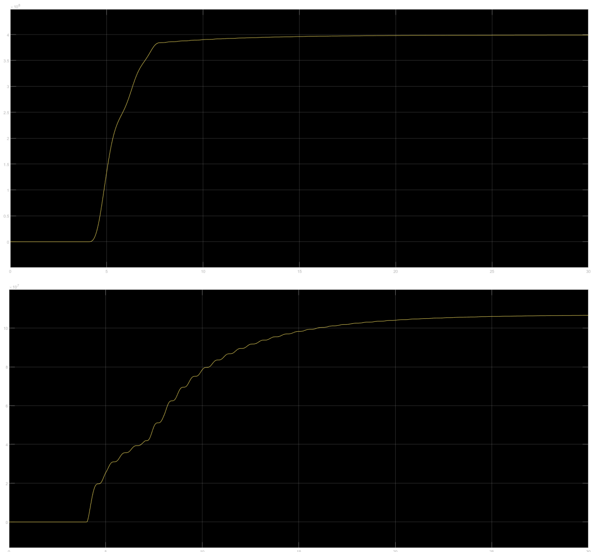


Figure 20: RMSE while implementing Explicit Control

2 Problem 2

In this problem a Hybrid MPC model is simulated. The baseline is a Linear MPC in reality. For the purose of this simulation we have made use of a switch that changes the dynamic used to create the output if a certain condition is satisfied. The general configuration is as follows in figure 21:

The model was defined in the workspace using the following command lines:

```
A1 = [0 2 ; -2 -2];
B1 = [0;2];
A2 = [0 4; -4 -4]
B2 = [0;4]
C = [1 0]
```

Figure 23 shows our state and input weights, and figure 22 shows the input and output constraints.

The reference signal is defined as $20sin(\frac{2\pi}{20}t)$. Figure 24 plots its reference

Figure 25 shows the output trajectory y(t), figure 26 shows control input u(t), figure 27 shows the Mode Switch Signal S(t) and figure 28 shows the reference alongside the output trajectory

The figures above show with switch changing dynamics roughly with a second delay the system experiences distortion in tracking the sinusoid system. It is also evident that there is a similarity between the Input signal and output signal, meaning that there is a distortion in the input signal as well. This mean that the MPC Controller, changes the the input signal based on the dynamic change it predicts in the future and can compensate for that in the output to obtain a more clear reference tracking in the output.

About the constraints that has been asked in the question, since the system was described in a way that did not reach the upper and lower bounds defined by the constraints, the output of the system does not differ for the "With Constraint" and the "Without Constraint" case.

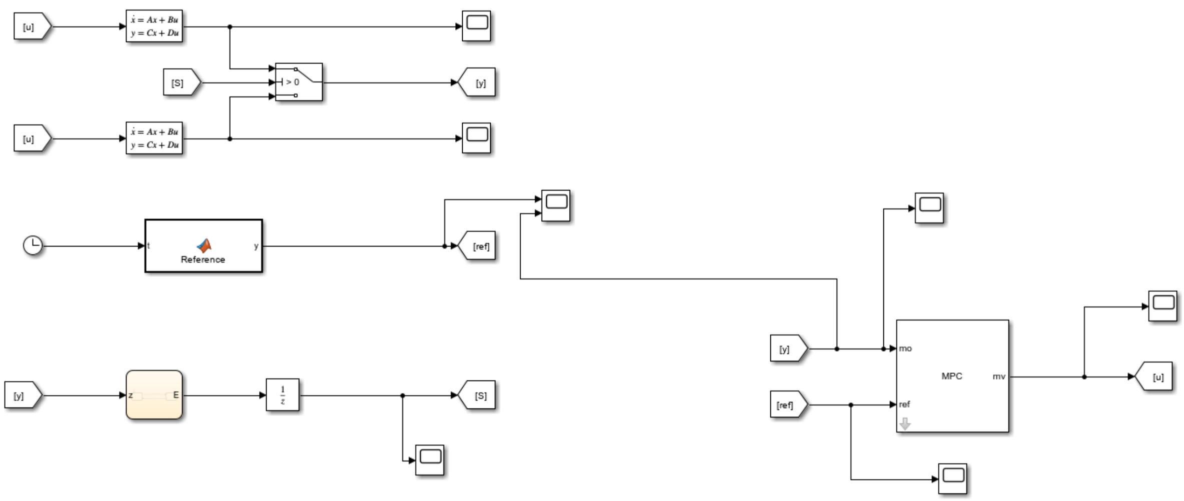


Figure 21: The Hybrid MPC model

Channel	Type	Min	Max	RateMin	RateMax
▼ **Inputs**					
u(1)	*MV*	-30	30	-Inf	Inf
▼ **Outputs**					
y(1)	*MO*	-30	30		

Figure 22: Input and Output Constraints

Input Weights (dimensionless)

	Channel	Type	Weight	Rate Weight	Target
1	u(1)	MV	0.1	0.1	nominal

Output Weights (dimensionless)

	Channel	Type	Weight
1	y(1)	MO	5

Figure 23: State and input weights

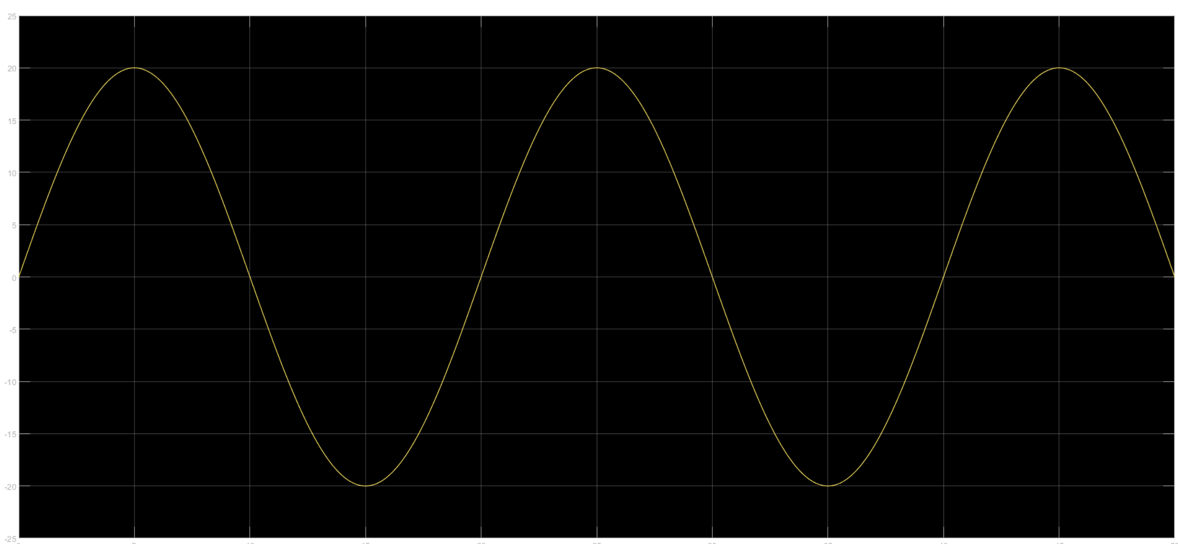


Figure 24: Sine Reference

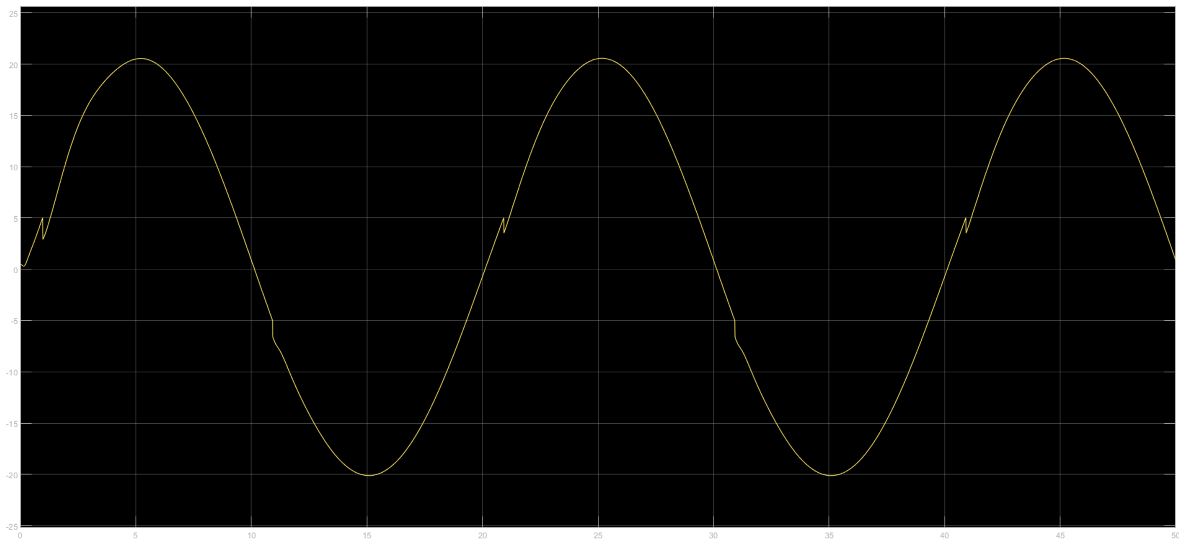


Figure 25: Output Trajectory

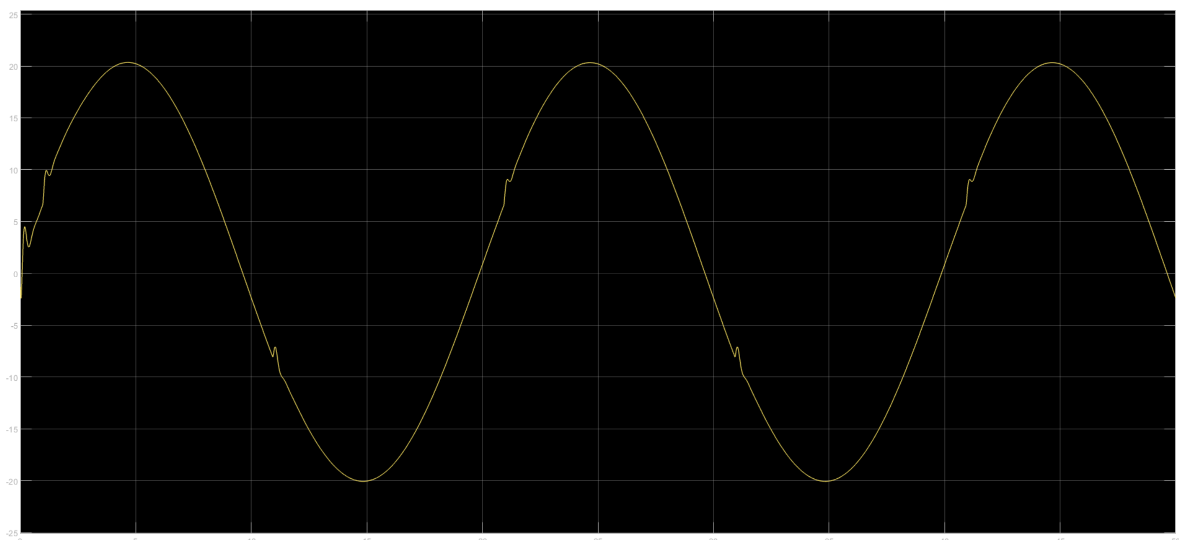


Figure 26: The Control Input in Hybrid

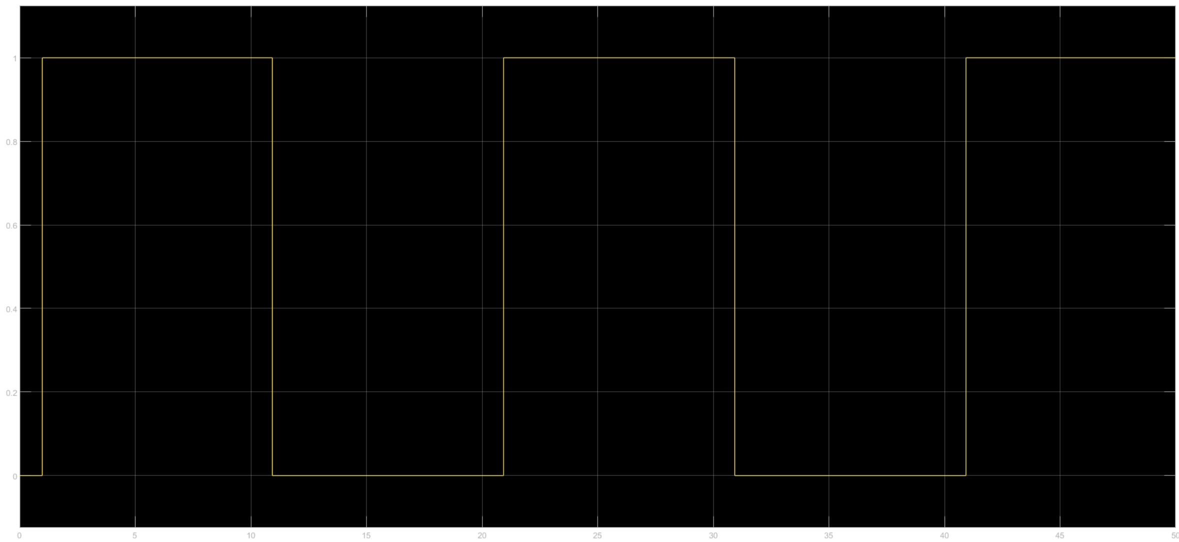


Figure 27: Mode Switch figure

Figure 28: Reference alongside Output with respect to time(t)