Model Predictive Controller

1st Karthik Sathyanarayanan

I. Part-A

A. Task A.1

The prediction of future system output can be given by :

$$\vec{x_k} = G\vec{u_k} + Hx_k$$

In this case we take the prediction horizon of N = 3. When we plug it into the equation for prediction, we get :

$$\begin{bmatrix} x_k0 \\ x_k1 \\ x_k2 \\ x_k3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 \\ AB & B & 0 & 0 \\ A^2B & AB & B & 0 \end{bmatrix} \begin{bmatrix} u_k0 \\ u_k1 \\ u_k2 \\ u_k3 \end{bmatrix} + \begin{bmatrix} 1 \\ A \\ A^2 \\ A^3 \end{bmatrix} *$$

x$_k$0

$where\ A = B = 1.$

B. Task A.2

The following equation demonstrate the minimised solution for unconstrained MPC design .

$$u_k0 = -(\bar{E} + G^T\bar{Q}G)^-1G^T\bar{Q}Hx_{t0}$$

K is given by the equation

$$K = -[1, 0...0](\bar{R} + G^T\bar{Q}G)^-1G^T\bar{Q}H$$

$$K = -\begin{bmatrix} \frac{\rho}{\rho^2+\rho} \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Simplifying further we get

$$K = -\frac{\rho}{\rho^2 + \rho}$$

$$u_{t0} = -\frac{\rho}{\rho^2 + \rho}x_{t0}$$

C. Task A.3

Closed loop pole is given by

$$\frac{\frac{1}{z-1}}{1 + \frac{K}{z-1}}$$

Setting the denominator to zero , we get

$$z = \frac{\rho}{\rho^2 + \rho} + 1$$

II. Part-B

A. Task B.1

The problem can be solved by standard finite horizon LQR controller by solving the algebraic riccati solution manually in a loop for 50 time steps.The initial condition is assigned and by dynamic programming , the LQR design is created .

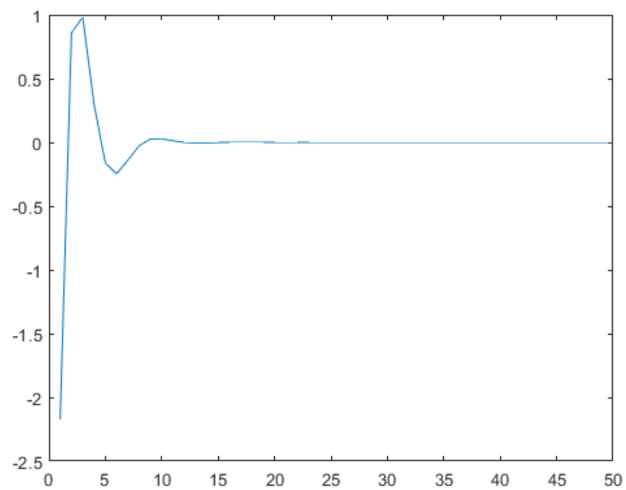

Fig. 1. Optimal input as function of time.

The above figure illustrates the optimal time input as a function of time .

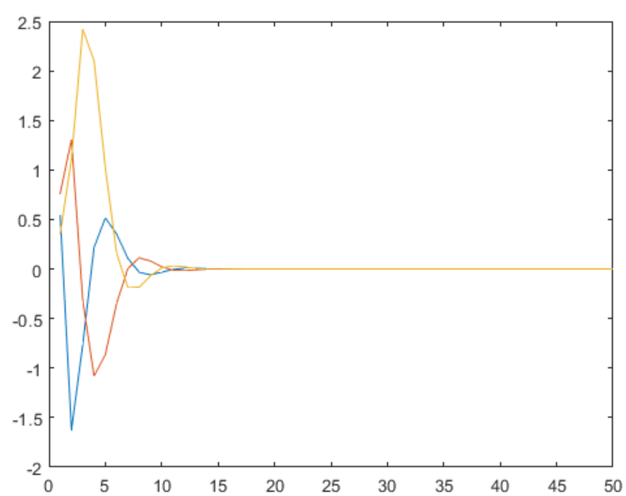

Fig. 2. Resulting states as a function of time.

The above figure illustrates resulting states as a function of time .

B. Task B.2

The MPC model is made without any constraints applied. The design is made by first making the G and H matrices by applying the prediction horizon of N = 5. Based on the

prediction the $\bar{u_{t0}}$ is applied as the control input and repeated for for 50 time steps.
The figure below illustrates the input signal applied over 50 time steps.

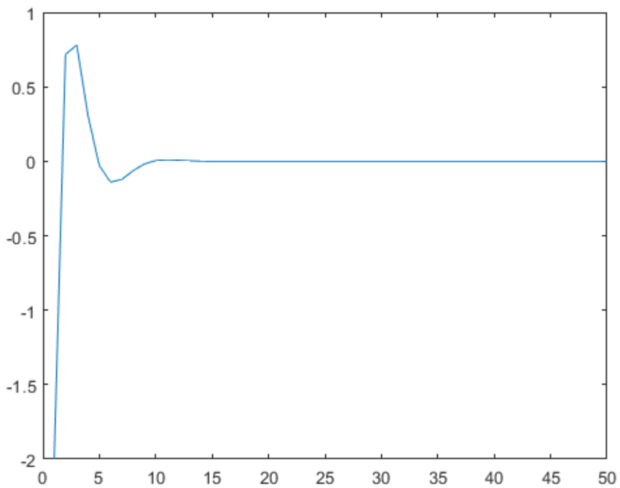

Fig. 3. Optimal input as a function of time

It can be seen that there is a variation in the way that the input signal reaches steady state . The figure below illustrates the resulting states vary before they reach a steady state as a function of time . We can see that the resulting states gain different values compared to the resulting states in LQR design.

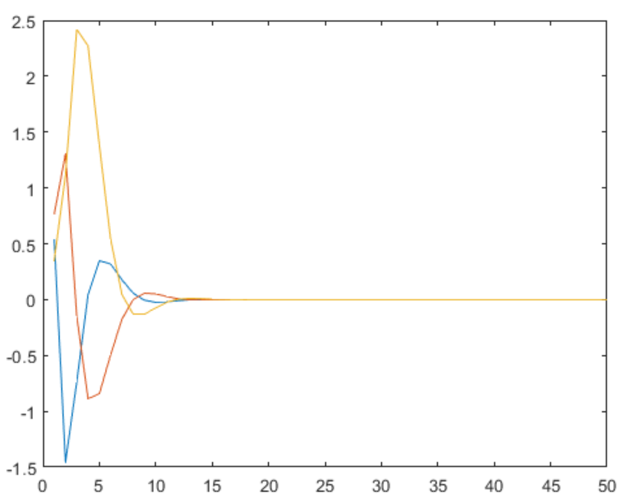

Fig. 4. Resulting states as a function of time.

C. Task B.3

The model predictive controller is now applied through constraints. We cannot find the u^*_{t0} analytically by minimising the cost function directly . We use the function quadprog in MATLAB to get the value after minimising the equation $\frac{1}{2}x^TQx+c^Tx$ The above figure illustrates that the states follow

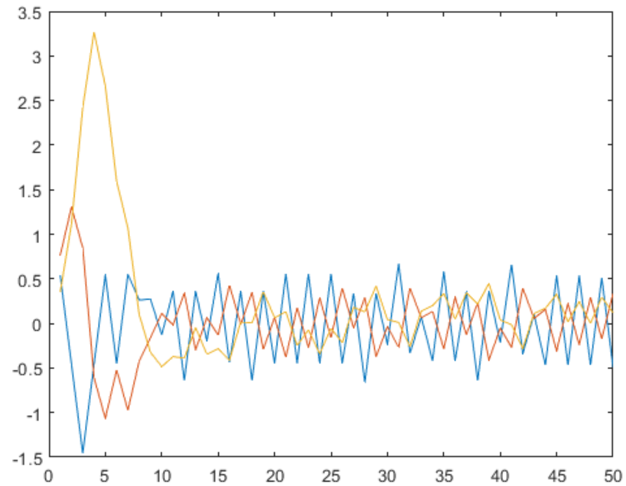

Fig. 5. Resulting states as a function of time.

a similar path as the earlier ones. The jitter in the graph for the resulting states to reach a steady state might be because of the quadprog functions of matlab not being able to adapt the symmetric matrix G in its internal equations. This jitter is also visible in the input function $\bar{u_t}$ which jitters from -1 to +1 obeying the constraints that wee fed in the matlab code. From the figures plotted above we can see that the LQR and in the unconstrained MPC design, the input signal did start from the value of -1.5 and -2 respectively which were out of the ranges of the input constraints . As we can see from the

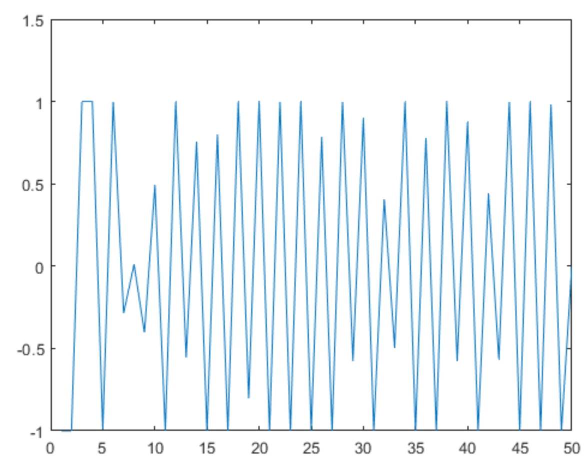

Fig. 6. Optimal input as a function of time.

figure , the optimal input even though jittery tries to reach the value of zero at 10 second mark but due to the Matlab toolbox internal calculations, the input signal oscillates rapidly.