Excellent - 7 out of 10

ELEC6228 Model Predictive Control

Oliver Schamp (30305861)

remember to define acronyms on first use.

Abstract okay but a little bit vague about contents...

A proper intro and a conclusion (and perhaps refs would fit the IEEE form)

ry physical system has some form of constraint to input, state and transition between inputs. Model ol is used to model these constraints? which are not an infinite/finite LQR solution.

I. A.1

system for a prediction horizon of N=3:

$$x_{k+1} = x_k + u_k$$
$$y_k = x_k$$
$$J = \sum_{t=t0}^{t0+N-1} {y_t}^2 + \rho {u_t}^2$$

he state is a single value and not a vector, we = 1, B = 1, C=1, R=ρ, Q = 1, $\mathcal{R} = \rho * I(N), \mathcal{Q} = \mathrm{I(N)}$. Representing the state equation in the form of $G\vec{u_k} + Hx_k$:

need to define new matrices and vectors when they appear

$$\begin{bmatrix} x_k \\ x_{k+1} \\ x_{k+2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x_k$$

II. A.2

Ignoring the constraint on the control signal, for a predictive horizon of N=2 solve the design problem. This is done by minimizing the gradient of the cost function displayed above. Predictive horizon N=2 of form $G\vec{u_k} + Hx_k$ gives:

$$\begin{bmatrix} x_k \\ x_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_k$$

Minimise the cost:

$$\min_{\vec{u_k}} (G\vec{u_k} + Hx_k)^T (G\vec{u_k} + Hx_k) + \rho u_k^2$$
$$\vec{u_k} = -([\rho * I] + G^T G)^{-1} G^T H x_k$$
$$\begin{bmatrix} u_k \\ u_{k+1} \end{bmatrix} = -\left(\begin{bmatrix} \rho & 0 \\ 0 & \rho \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_k$$
$$\begin{bmatrix} u_k \\ u_{k+1} \end{bmatrix} = -\frac{1}{\rho(\rho+1)} \begin{bmatrix} \rho \\ 0 \end{bmatrix} x_k$$
$$\begin{bmatrix} u_k \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\rho+1} \\ 0 \end{bmatrix} x_k$$

This allows a state feedback controller expression to be written as follows:

$$u_k = -\frac{1}{\rho+1} x_k$$

III. A.3

Closed loop pole of the system:

$$Eig(zI - (A - BK))$$

Find A-BK:

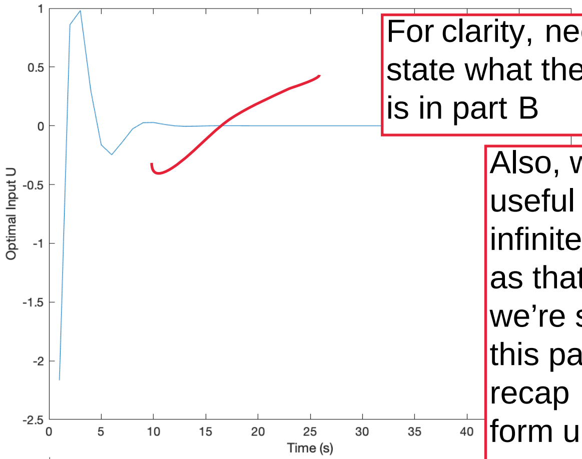
$$CLP = 1 \overset{-}{+} \frac{1}{\rho+1} = \frac{\rho+2}{\rho+1}$$

As this is the only value present, it can be concluded that the system has a closed loop pole at this value.

IV. B.1

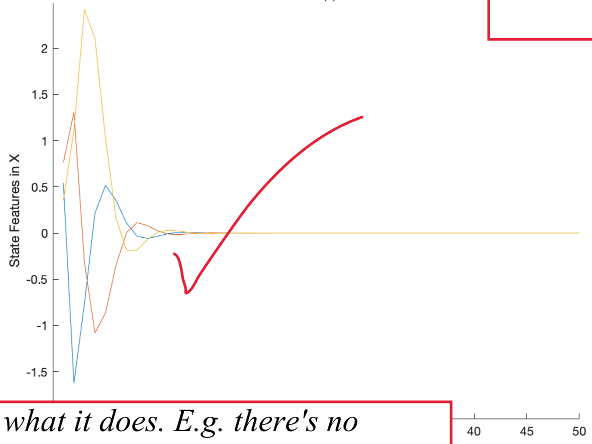
For this problem, the values of A, B, C and D were given along with the transfer function dependent on A and the cost function $\rho \sum_{t=0}^{N-1} {u_t}^2 + \sum_{t=0}^{N} {y_t}^2$ and the outp function $y_k = Cx_k, Q_f = C^T C$. From this value of Q previous P matrices were calculated from the recurrence relationship in the Riccati Equation.

we're solving the infinite horizon case



For clarity, need to state what the system is in part B

Also, would be useful to state infinite horizon cost as that's what we're solving in this part. And recap general soln form uk = K xk



Follows well but does not give any general detail on MPC or what it does. E.g. there's no general ss model, so reference to A, B, C doesn't follow. The marks scheme has "professional technical writing" as a requirement, so throughout the report everything should be defined logically and clearly, including the MPC problem you are solving, and all related vectors and all matrices. You don't have to give derivations, but it should make sense to a general reader..

To calculate the cost, the cost function for a standard finite horizon LQR problem was used:

$$J = V_0(x_0) = x_0^T P x_0$$

The cost of this method was returned as 19.3759.

V. B.2

Solving the problem using Model Predictive Control with a prediction horizon of N=5 in an unconstrained method. The values for G, H, Q and R:

For clarity, need to state the receding horizon cost function we're solving here

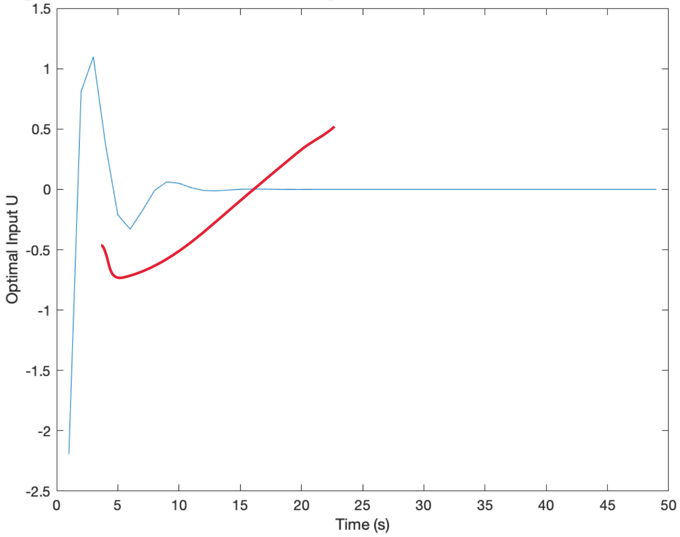
$$\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ B & 0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ A^3B & \cdots & B & 0 \end{bmatrix}, H = \begin{bmatrix} I \\ A \\ \vdots \\ A^4 \end{bmatrix}$$

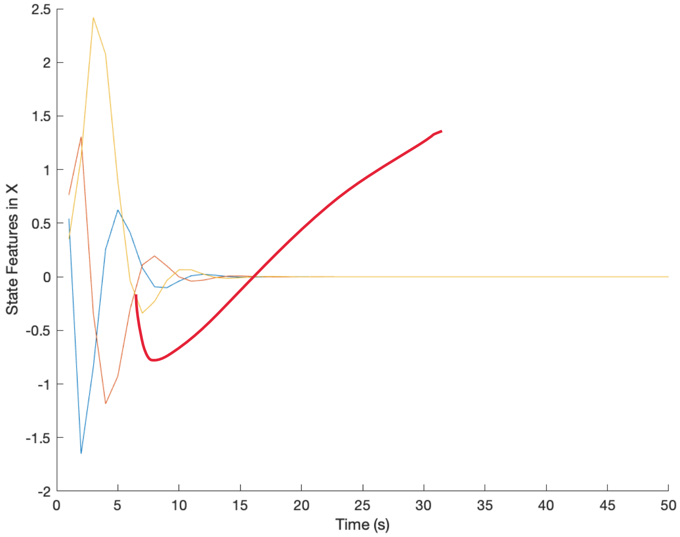
$$Q = C^T C, R = \rho * I(5)$$

derivative of the cost function, K was calculated. $\mathcal{R}$ and $\mathcal{Q}$ are the R and Q matrices concatenated by formation of the I matrix: for example, $\mathcal{Q} = \begin{bmatrix} Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q \end{bmatrix}$.

$$K = -\begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} (\mathcal{R} + G^T \mathcal{Q} G)^{-1} G^T \mathcal{Q} H$$

Using K to calculate the optimal input $u_k^* = K x_k$ the optimal inputs and the states for 50 steps were obtained.





Less performant than LQR-cost found iteratively to be 28.7036.

VI. B.3

If a constraint of $|u_k| \leq 1$ is added, both previously would be in violation of that constra inputs are -2.166 (B1) and -2.194 (B2).

Ideally we need to give the basic objective function J for constrained MPC too (with constraints), and a little more discussion/ explanation how it works

With the input constraint of $|u_k| \leq 1$, the N=5

$$\begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \leq \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ u_{k+3} \\ u_{k+4} \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = (-1)^{\rightarrow} \leq u_k^{\rightarrow}$$

This can be rewritten as:

$$\begin{bmatrix} I(5) \\ -I(5) \end{bmatrix} u_k^{\rightarrow} \leq \begin{bmatrix} 1^{\rightarrow} \\ 1^{\rightarrow} \end{bmatrix}$$

This gives the form of $F u_k^{\rightarrow} \leq b$.

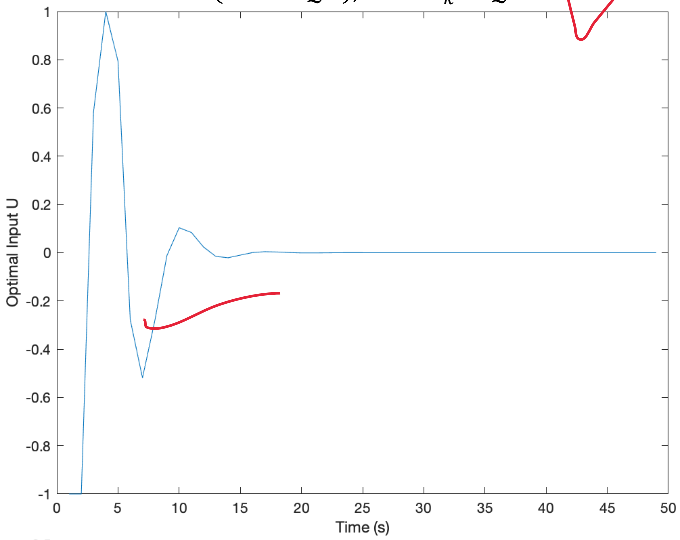
Expanding the cost equation $J(u_k^{\rightarrow})$:

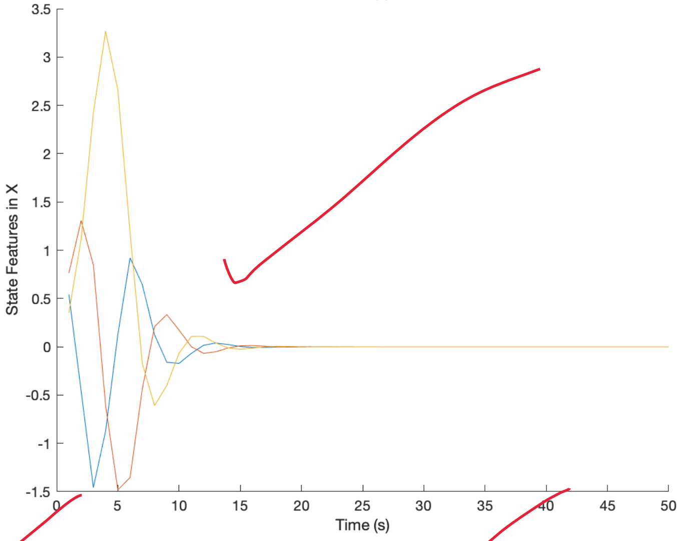
$$J(u_k^{\rightarrow}) = u_k^{\rightarrow T} (\mathcal{R} + G^T \mathcal{Q} G) u_k^{\rightarrow} + 2 x_k^T H^T \mathcal{Q} G u_k^{\rightarrow} + x_k^T H^T \mathcal{Q} H x_k$$

Remove the constant term $x_k^T H^T \mathcal{Q} H x_k$ and MATLAB can solve the constrained problem by quadratic programming, as the method becomes a minimization of form:

$$\min_{u_k^{\rightarrow}} \frac{1}{2} u_k^{\rightarrow T} Z u_k^{\rightarrow} + Y^T u_k^{\rightarrow}$$

$$Z = (\mathcal{R} + G^T \mathcal{Q} G), Y = 2 x_k^T H^T \mathcal{Q} G$$





In this design, the constraint $|u_k| \leq 1$ is satisfied.

This method takes longer to converge on a stable state due to the power constraints on the input limiting the maximum effect a single step can have on the system. Because of this cost was found iteratively to be 43.9133, which is the highest value from all the methods.

are the states higher too?

your costs are different to what I was expecting. Perhaps check if you have time

Good discussion – I think it includes most aspects. well done on getting these results