Linear Model Predictive Control

1: Create augmented model with integration

State space model – This is what you begin with.

$$x_m(k+1) = A_m x_m(k) + B_m u(k)$$
$$y(k) = C_m x_m(k)$$

Where:

$$A_m \in \Re^{nxn}$$
$$B_m \in \Re^{nxm}$$
$$C_m \in \Re^{qxn}$$
$$o_m \in 0^{qxn}$$
$$x_m \in \Re^{n}$$
$$y \in \Re^{q}$$
$$u \in \Re^{m}$$

Augmented state space model with internal integration

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & I_{qxq} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k)$$
$$y(k) = \begin{bmatrix} o_m & I_{qxq} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

Where:

$$\Delta u(k) = u(k) - u(k-1)$$
$$\Delta x_m(k) = x_m(k) - x_m(k-1)$$
$$\Delta x_m(k+1) = x_m(k+1) - x_m(k)$$

2: Create the prediction matrecies

Prediction equation.

$$Y = Fx(k_i) + \Phi \Delta U$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}, \Phi = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \cdots & CA^{N_p-N_c}B \end{bmatrix}$$

Where:

$$A = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & I_{qxq} \end{bmatrix}$$
$$B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}$$
$$C = \begin{bmatrix} o_m & I_{qxq} \end{bmatrix}$$
$$x(k_i) = \begin{bmatrix} \Delta x_m(k_i) \\ y(k_i) \end{bmatrix} (Current\, state)$$
$$N_p \in \Re\, (Prediction\, horizon)$$
$$N_c \in \Re\, (Control\, horizon)$$
$$k_i \in \Re\, (Sampling\, step)$$

Future trajectories of inputs and outputs:

$$\Delta U = \begin{bmatrix} \Delta u(k_i) & \Delta u(k_i+1) & \Delta u(k_i+2) & \cdots & \Delta u(k_i+N_c-1) \end{bmatrix}^T$$
$$Y = \begin{bmatrix} y(k_i+1|k_i) & y(k_i+2|k_i) & y(k_i+3|k_i) & \cdots & y(k_i+N_p|k_i) \end{bmatrix}^T$$

Where $y(k_i+1|k_i)$ is interprented as ”Predicted output variable at k_i+1 with given current plant information position k_i ”.

3: Open loop MPC

Minimize the cost function and find the best input trajectories.

$$J=(r(k_i)-Y)^T(r(k_i)-Y)+\Delta U^T\bar{R}\Delta U$$

This can be reformulated as:

$$J=(r(k_i)-Fx(k_i))^T(r(k_i)-Fx(k_i))-2\Delta U^T\Phi^T(r(k_i)-Fx(k_i))+\Delta U^T(\Phi^T\Phi+\bar{R})\Delta U$$

And to minimize the derivative of J :

$$\frac{\partial J}{\partial \Delta U}=-2\Phi^T(r(k_i)-Fx(k_i))+2(\Phi^T\Phi+\bar{R})\Delta U=0$$

And find the optimal input trajectories ΔU when $J=0$:

$$\Delta U=(\Phi^T\Phi+\bar{R})^{-1}\Phi^T(r(k_i)-Fx(k_i))$$

Where:

$$\begin{aligned}
&\bar{R}=r_\omega I_{N_c x N_c}\\
&r_\omega\geq 0\in\Re(\textit{Tuning parameter})\\
&r(k_i)\in\Re^{N_p}(\textit{Set}-\textit{point vector})
\end{aligned}$$

4: Closed loop MPC

Find the static control law.

$$\Delta u(k_i)=K_y r(k_i)-K_{mpc}x(k_i)$$

And create the closed loop MPC controller

$$\begin{aligned}
&x(k_i+1)=[(A-BK_{mpc})]x(k_i)+[BK_y]r(k_i)\\
&y(k_i)=Cx(k_i)
\end{aligned}$$

Where:

$$\begin{aligned}
&K_y=[\overbrace{I_{mxm}\quad I_{mxm}\times 0\quad I_{mxm}\times 0\quad\cdots\quad I_{mxm}\times 0}^{N_c}](\Phi^T\Phi+\bar{R})^{-1}\Phi^T\bar{R}_s\\
&K_{mpc}=[\overbrace{I_{mxm}\quad I_{mxm}\times 0\quad I_{mxm}\times 0\quad\cdots\quad I_{mxm}\times 0}^{N_c}](\Phi^T\Phi+\bar{R})^{-1}\Phi^T F\\
&\bar{R}_s=[\overbrace{1\quad 1\quad 1\quad 1\quad\cdots\quad 1}^{N_p}]^T
\end{aligned}$$

5: Closed loop MPC with kalman filter

Use mpcreg.m and get the kalman gain matrix from lqe.m function.