CL686 Course Project: Generic Description
Closed Loop Simulations using
MPC Algorithms based on State Augmentation Approach

1 System Dynamic Simulation

Let the nonlinear system assigned to you be described as follows

$$\begin{aligned} \frac{d\mathcal{X}}{dt} &= \mathbf{f}\left(\mathcal{X},\mathcal{U},\mathcal{D}\right) && (1) \\ \mathcal{Y} &= \mathbf{g}(\mathcal{X}) && (2) \\ \mathcal{X} &\in R^n, \mathcal{U}\in R^m, \mathcal{D}\in R^d, \mathcal{Y}\in R^r \end{aligned}$$

and let $(\mathcal{X}_s,\mathcal{U}_s,\mathcal{D}_s)$ represent the specified equilibrium operating point. Dynamics of the system is simulated the with the specified initial condition $\mathcal{X}(0)$ and with piecewise constant inputs

$$\mathcal{U}\left(t\right) = \mathcal{U}_s + \mathbf{u}(k) \quad \text{for } kT \leq t < (k+1)T \tag{3}$$

$$\mathcal{D}\left(t\right) = \mathcal{D}_s + \mathbf{d}(k) \quad \text{for } kT \leq t < (k+1)T \tag{4}$$

where $\mathbf{u}(k)$ represents the controller output and $\mathbf{d}(k)$ is a zero mean Gaussian white noise sequence with covariance matrix $\mathbf{Q}_d$. The measurements available from the system at sampling interval T are corrupted with noise

$$\mathcal{Y}(k) = \mathbf{C}\mathcal{X}(k) + \mathbf{v}(k) \tag{5}$$

where $\mathbf{v}(k)$ is a zero mean Gaussian white noise sequence with covariance matrix $\mathbf{R}$.

2 Control Relevant Perturbation Model

Consider the discrete time linear model obtained through linearization of the mechanistic model in the neighborhood of $\left(\overline{\mathcal{X}},\overline{\mathcal{U}},\overline{\mathcal{D}}\right)$

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}_u\mathbf{u}(k) + \mathbf{\Gamma}_d\mathbf{d}(k) && (6) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\begin{bmatrix} \mathbf{u}(k) \\ \mathbf{d}(k) \end{bmatrix} + \mathbf{v}(k) && (7) \\ \mathbf{x}(k) &= \mathcal{X}(k) - \mathcal{X}_s, \mathbf{u}(k) = \mathcal{U}(k) - \mathcal{U}_s, && (8) \\ \mathbf{d} &= \mathcal{D}\left(k\right) - \mathcal{D}_s, \mathbf{y}(k) = \mathcal{Y}(k) - \mathbf{C}\mathcal{X}_s && (9) \end{aligned}$$

where $\mathbf{d}(k)$ and $\mathbf{v}(k)$ are a zero mean Gaussian white noise sequences with covariance matrices $\mathbf{Q}_d$ and $\mathbf{R}$,respectively. Here, $\mathbf{D}$ is a **null matrix** of dimension $r \times (m+d)$. Also, define covariance matrix

$$\mathbf{N} = E\left[\mathbf{d}(k)\mathbf{v}(k)^T\right] = [\mathbf{0}]_{d\times r}$$

which is a null matrix since disturbance $\mathbf{d}(k)$ and measurement noise $\mathbf{v}(k)$ are uncorrelated.

3 Kalman Predictor for Augmented State Space Model

When bias states are introduced in the manipulated input, we have

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}_u\left[\mathbf{u}(k) + \boldsymbol{\beta}(k)\right] + \mathbf{\Gamma}_d\mathbf{d}(k) && (10)\\ \boldsymbol{\beta}(k+1) &= \boldsymbol{\beta}(k) + \mathbf{w}_\beta(k) && (11)\\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) && (12) \end{aligned}$$

Thus, we can define augmented model

$$\begin{aligned} \mathbf{x}_a(k+1) &= \mathbf{\Phi}_a\mathbf{x}_a(k) + \mathbf{\Gamma}_{ua}\mathbf{u}(k) + \mathbf{\Gamma}_{da}\mathbf{d}_a(k) && (13)\\ \mathbf{y}(k) &= \mathbf{C}_a\mathbf{x}_a(k) + \mathbf{D}_a\begin{bmatrix}\mathbf{u}(k)\\ \mathbf{d}_a(k)\end{bmatrix} + \mathbf{v}(k) && (14) \end{aligned}$$

where

$$\begin{aligned} \mathbf{x}_a(k) &= \begin{bmatrix}\mathbf{x}(k)\\ \boldsymbol{\beta}(k)\end{bmatrix}_{(n+m)\times 1} \quad ; \quad \mathbf{d}_a(k) = \begin{bmatrix}\mathbf{d}(k)\\ \mathbf{w}_\beta(k)\end{bmatrix}_{(d+m)\times 1} && (15)\\ \mathbf{\Phi}_a &= \begin{bmatrix}\mathbf{\Phi} & \mathbf{\Gamma}_u\\ [\mathbf{0}] & \mathbf{I}_\beta\end{bmatrix}_{(n+m)\times(n+m)} ; \ \mathbf{\Gamma}_{ua} = \begin{bmatrix}\mathbf{\Gamma}_u\\ \mathbf{0}\end{bmatrix}_{(n+m)\times m} ; \ \mathbf{\Gamma}_{da} = \begin{bmatrix}\mathbf{\Gamma}_d & [\mathbf{0}]\\ [\mathbf{0}] & \mathbf{I}_\beta\end{bmatrix}_{(d+m)\times(d+m)} && (16)\\ \mathbf{C}_a &= \begin{bmatrix}\mathbf{C} & [\mathbf{0}]\end{bmatrix}_{r\times(n+m)} \quad ; \quad \mathbf{D}_a = [\mathbf{0}]_{r\times(m+d+m)} && (17) \end{aligned}$$

$$\begin{aligned} \mathbf{Q}_a &= E\left[\mathbf{d}_a(k)\mathbf{d}_a(k)^T\right] = \begin{bmatrix}\mathbf{Q}_d & [\mathbf{0}]\\ [\mathbf{0}] & \mathbf{Q}_\beta\end{bmatrix}_{(d+m)\times(d+m)} && (18)\\ \mathbf{N}_a &= E\left[\mathbf{d}_a(k)\mathbf{v}(k)^T\right] = \begin{bmatrix}[\mathbf{0}]\\ [\mathbf{0}]\end{bmatrix}_{(d+m)\times r} && (19)\\ \mathbf{R}_a &= E\left[\mathbf{v}(k)\mathbf{v}(k)^T\right] = \mathbf{R} && (20) \end{aligned}$$

This augmented model can be used to develop Kalman predictor of the form

$$\begin{aligned} \mathbf{e}_a(k) &= \mathbf{y}(k) - \mathbf{C}_a\widehat{\mathbf{x}}_a(k) && (21)\\ \widehat{\mathbf{x}}_a(k+1) &= \mathbf{\Phi}_a\widehat{\mathbf{x}}_a(k) + \mathbf{\Gamma}_{ua}\mathbf{u}(k) + \mathbf{L}_a\mathbf{e}_a(k) && (22) \end{aligned}$$

where the steady state Kalman gain is obtained by solving the corresponding steady state Riccati equations

$$\mathbf{L}_a = \left[\mathbf{\Phi}_a \mathbf{P}_{a\infty} \mathbf{C}_\mathbf{a}^\mathbf{T} + \mathbf{N}\right] \left[\mathbf{C}_a \mathbf{P}_{a\infty} \mathbf{C}_\mathbf{a}^\mathbf{T} + \mathbf{R}_a\right]^{-1} \tag{23}$$

$$\mathbf{P}_{a\infty} = \mathbf{\Phi}_a \mathbf{P}_{a\infty} \mathbf{\Phi}_a^T + \mathbf{\Gamma}_{da} \mathbf{Q}_a \mathbf{\Gamma}_{da}^T - \mathbf{L}_a \left[\mathbf{C}_a \mathbf{P}_{a\infty} \mathbf{C}_\mathbf{a}^\mathbf{T} + \mathbf{R}_a\right] \mathbf{L}_a^T \tag{24}$$

Solution of the ARE can be found using Matlab Control System Toolbox function *kalman* as follows

1. Step 1: Create a state space object using ss command

   $dmod = ss(\ \mathbf{\Phi}_a,\ \begin{bmatrix} \mathbf{\Gamma}_{ua} & \mathbf{\Gamma}_{da} \end{bmatrix},\ \mathbf{C}_a,\ \mathbf{D}_a,\ T\ )$

   where T represents sampling interval.

2. Call Matlab function *kalman*

   $[KEST,\ \mathbf{L}_a, \mathbf{P}_{a\infty}] = kalman(dmod,\ \mathbf{Q}_a,\ \mathbf{R}_a, \mathbf{N}_a)\ ;$

Implementation of augmented state estimator

$$\mathbf{y}(k) = \mathcal{Y}(k) - \mathcal{Y}_s \tag{25}$$

$$\mathbf{e}(k) = \mathbf{y}(k) - \mathbf{C}\widehat{\mathbf{x}}(k) \tag{26}$$

$$\widehat{\mathbf{x}}_a(k+1) = \mathbf{\Phi}_a \widehat{\mathbf{x}}(k) + \mathbf{\Gamma}_{ua}\mathbf{u}(k) + \mathbf{L}_a \mathbf{e}(k) \tag{27}$$

yields state sequence

$$\widehat{\mathbf{x}}_a(k) = \begin{bmatrix} \widehat{\mathbf{x}}(k) \\ \widehat{\boldsymbol{\beta}}(k) \end{bmatrix} \tag{28}$$

for $k = 1, 2, ...$which can be used to implement MPC. Further, the target steady state $\mathbf{x}_s(k)$ and target input $\mathbf{u}_s(k)$ are computed as follows

$$\mathbf{u}_s(k) = \mathbf{K}_u^{-1}\mathbf{r}(k) - \widehat{\boldsymbol{\beta}}(k) \tag{29}$$

$$\mathbf{x}_s(k) = (\mathbf{I} - \mathbf{\Phi})^{-1}\, \Gamma_u \mathbf{K}_u^{-1} \mathbf{r}(k) \tag{30}$$

$$\mathbf{K}_u = \mathbf{C}\,(\mathbf{I} - \mathbf{\Phi})^{-1}\, \Gamma_u \tag{31}$$

where $\mathbf{r}(k)$ denotes desired setpoint at instant k.

4 Linear Model Predictive Control

Let us assume that $\mathbf{r}(k)$ denotes desired setpoint at instant k and let $(\mathbf{x}_s(k),\ \mathbf{u}_s(k))$ represent the corresponding target steady state and target input, respectively, computed using $\mathbf{r}(k)$. The model predictive control problem at the sampling instant k is defined as a constrained optimization problem whereby the future manipulated input moves

$$\mathcal{U}_f = \{\mathbf{u}(k+j|k):\ j = 0, 1, ...q-1\} \tag{32}$$

are determined by minimizing a cost function defined over prediction horizon p. Here, q is known as the control horizon. Note that use of control horizon $q < p$ implies inclusion of the following constraints on the future manipulated inputs

$$\mathbf{u}(k+q|k) = \mathbf{u}(k+q+1|k) = ... = \mathbf{u}(k+p-1|k) = \mathbf{u}(k+q-1|k) \tag{33}$$

Let $\mathbf{W}_x$,$\mathbf{W}_u$, $\mathbf{W}_{\Delta u}$ and $\mathbf{W}_x$ represent +ve definite matrices. Four different constrained optimization based MPC schemes are described here using these matrices.

MPC-1

$$\underset{\mathcal{U}_f}{\arg Min}\ J = \sum_{j=1}^{p} \boldsymbol{\varepsilon}(k+j|k)^T \mathbf{W}_x \boldsymbol{\varepsilon}(k+j|k) + \sum_{j=0}^{q-1} \delta\mathbf{u}(k+j|k)^T \mathbf{W}_u \delta\mathbf{u}(k+j|k) \tag{34}$$

$$\begin{aligned} \boldsymbol{\varepsilon}(k+j|k) &= \mathbf{x}_s(k) - \widehat{\mathbf{z}}(k+j) \quad \text{for} \quad j = 1, 2, ...p \\ \mathbf{x}_s(k) &: \text{computed using (??)} \end{aligned} \tag{35}$$

$$\begin{aligned} \delta\mathbf{u}(k+j|k) &= \mathbf{u}(k+j|k) - \mathbf{u}_s(k) \quad \text{for} \quad j = 1, ...q-1 \\ \mathbf{u}_s(k) &: \text{computed using (??)} \end{aligned} \tag{36}$$

Subject to

$$\widehat{\mathbf{z}}(k+j+1) = \boldsymbol{\Phi}\widehat{\mathbf{z}}(k+j) + \boldsymbol{\Gamma}_u\left(\mathbf{u}(k+j|k) + \widehat{\boldsymbol{\beta}}(k)\right) \tag{37}$$

$$\text{for}\ \ j = 1, 2, ...p. \text{ with } \widehat{\mathbf{z}}(k) = \widehat{\mathbf{x}}(k) \tag{38}$$

$$\widehat{\mathbf{x}}(k) : \text{computed using eq. (22)}$$

$$\begin{aligned} \mathbf{u}_L &\le \mathbf{u}(k+j|k) \le \mathbf{u}_H \\ j &= 0, 1, 2, ...q-1 \end{aligned} \tag{39}$$

MPC-2

$$J = \sum_{j=1}^{p} \boldsymbol{\varepsilon}(k+j|k)^T \mathbf{W}_x \boldsymbol{\varepsilon}(k+j|k) + \sum_{j=0}^{q-1} \Delta\mathbf{u}(k+j|k)^T \mathbf{W}_{\Delta u} \Delta\mathbf{u}(k+j|k)$$

$$\begin{aligned} \boldsymbol{\varepsilon}(k+j|k) &= \mathbf{x}_s(k) - \widehat{\mathbf{z}}(k+j) \quad \text{for} \quad j = 1, 2, ...p & (40) \\ \mathbf{x}_s(k) &: \text{computed using } (\mathbf{??}) \end{aligned}$$

$$\begin{aligned} \Delta\mathbf{u}(k+j|k) &= \mathbf{u}(k+j|k) - \mathbf{u}(k+j-1|k) \quad \text{for} \quad j = 1, ...q-1 & (41) \\ \Delta\mathbf{u}(k|k) &= \mathbf{u}(k|k) - \mathbf{u}(k-1) & (42) \end{aligned}$$

Subject to

$$\begin{aligned} \widehat{\mathbf{z}}(k+j+1) &= \boldsymbol{\Phi}\widehat{\mathbf{z}}(k+j) + \boldsymbol{\Gamma}_u \left(\mathbf{u}(k+j|k) + \widehat{\boldsymbol{\beta}}(k)\right) & (43) \\ \text{for} \quad j &= 1, 2, ...p. \text{ with } \widehat{\mathbf{z}}(k) = \widehat{\mathbf{x}}(k) & (44) \\ \widehat{\mathbf{x}}(k) &: \text{computed using eq. } (22) \\ \mathbf{e}_f(k) &: \text{computed using eq. } (\mathbf{??}) \end{aligned}$$

$$\begin{aligned} \mathbf{u}_L &\leq \mathbf{u}(k+j|k) \leq \mathbf{u}_H & (45) \\ j &= 0, 1, 2, ...q-1 \end{aligned}$$

MPC-3

$$\underset{\mathcal{U}_f}{\arg Min} \quad J = \sum_{j=1}^{p} \boldsymbol{\epsilon}(k+j|k)^T \mathbf{W}_y \boldsymbol{\epsilon}(k+j|k) + \sum_{j=0}^{q-1} \delta\mathbf{u}(k+j|k)^T \mathbf{W}_u \delta\mathbf{u}(k+j|k)$$

$$\boldsymbol{\epsilon}(k+j|k) = \mathbf{r}(k) - \widehat{\mathbf{y}}(k+j) \quad \text{for} \quad j = 1, 2, ...p \qquad (46)$$

$$\begin{aligned} \delta\mathbf{u}(k+j|k) &= \mathbf{u}(k+j|k) - \mathbf{u}_s(k) \quad \text{for} \quad j = 1, ...q-1 & (47) \\ \mathbf{u}_s(k) &: \text{computed using } (\mathbf{??}) \end{aligned}$$

Subject to

$$\begin{aligned} \widehat{\mathbf{z}}(k+j+1) &= \boldsymbol{\Phi}\widehat{\mathbf{z}}(k+j) + \boldsymbol{\Gamma}_u \left(\mathbf{u}(k+j|k) + \widehat{\boldsymbol{\beta}}(k)\right) & (48) \\ \widehat{\mathbf{y}}(k+j+1) &= \mathbf{C}\widehat{\mathbf{z}}(k+j+1) & (49) \\ \text{for} \quad j &= 1, 2, ...p \text{ with } \widehat{\mathbf{z}}(k) = \widehat{\mathbf{x}}(k) & (50) \\ \widehat{\mathbf{x}}(k) &: \text{computed using eq. } (22) \end{aligned}$$

$$\begin{aligned} \mathbf{u}_L &\leq \mathbf{u}(k+j|k) \leq \mathbf{u}_H \\ j &= 0, 1, 2, ...q-1 \end{aligned} \tag{51}$$

MPC-4

$$\underset{\mathcal{U}_f}{\arg Min} \; J = \sum_{j=1}^{p} \boldsymbol{\epsilon}(k+j|k)^T \mathbf{W}_y \boldsymbol{\epsilon}(k+j|k) + \sum_{j=0}^{q-1} \Delta\mathbf{u}(k+j|k)^T \mathbf{W}_{\Delta u} \Delta\mathbf{u}(k+j|k)$$

$$\boldsymbol{\epsilon}(k+j|k) = \mathbf{r}(k) - \widehat{\mathbf{y}}(k+j) \quad \text{for} \quad j = 1, 2, ...p \tag{52}$$

$$\Delta\mathbf{u}(k+j|k) = \mathbf{u}(k+j|k) - \mathbf{u}(k+j-1|k) \quad \text{for} \quad j = 1, ...q-1 \tag{53}$$

$$\Delta\mathbf{u}(k|k) = \mathbf{u}(k|k) - \mathbf{u}(k-1) \tag{54}$$

Subject to

$$\widehat{\mathbf{z}}(k+j+1) = \boldsymbol{\Phi}\widehat{\mathbf{z}}(k+j) + \boldsymbol{\Gamma}_u \left(\mathbf{u}(k+j|k) + \widehat{\boldsymbol{\beta}}(k)\right) \tag{55}$$

$$\widehat{\mathbf{y}}(k+j+1) = \mathbf{C}\widehat{\mathbf{z}}(k+j+1) \tag{56}$$

$$\text{for} \;\; j = 1, 2, ...p \text{ with } \widehat{\mathbf{z}}(k) = \widehat{\mathbf{x}}(k) \tag{57}$$

$$\widehat{\mathbf{x}}(k) \; : \; \text{computed using eq. (22)}$$

$$\begin{aligned} \mathbf{u}_L &\leq \mathbf{u}(k+j|k) \leq \mathbf{u}_H \\ j &= 0, 1, 2, ...q-1 \end{aligned} \tag{58}$$

In each case, after solving the optimization problem at instant k, only the first move $\mathbf{u}_{opt}(k|k)$ is implemented on the plant, i.e.

$$\mathbf{u}(k) = \mathbf{u}_{opt}(k|k) \tag{59}$$

and the optimization problem is reformulated at the next sampling instant based on the updated information from the plant.