(MPC) 1 Discrete-time MPC for Beginners / 1.1 Introduction

1. This lecture introduces the *basic ideas and terms* about model predictive control.
2. A single-input and single-output (SISO) state-space model with an *embedded integrator* is introduced, which is used in the design of discrete-time predictive controllers with integral action
3. The design of predictive control within one optimization window is examined for primitive study
4. The ideas of receding horizon control, and state feedback gain matrices, and the closed-loop configuration of the predictive control system are discussed
5. The results are extended to multi-input and multi-output (MIMO) systems
6. In a general framework of state-space design, an *observer* is needed in the implementation, and this is discussed
7. With a combination of estimated state variables and the predictive controller, the state estimate predictive control is presented including *separation principle*.

(MPC) 1.2 State-Space Models with Embedded Integrator

1. For simplicity, we begin our study by assuming that the underlying plant is a single-input and single-output (SISO) system (strictly proper, $D_m = 0$), described by:

$$
\begin{aligned}
x_m(k+1) &= A_m x_m(k) + B_m u(k) \\
y(k) &= C_m x_m(k)
\end{aligned}
$$

   where $x_m(k) \in \Re^{n_1}$, $u(k) \in \Re$, and $y(k) \in \Re$

2. For the integrator embedding, taking a difference operation gives us

$$
\begin{aligned}
x_m(k+1) - x_m(k) &= A_m[x_m(k) - x_m(k-1)] + B_m[u(k) - u(k-1)] \\
y(k+1) - y(k) &= C_m[x_m(k+1) - x_m(k)]
\end{aligned}
$$

   Let us denote the *difference of the state and control* variables

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

   where these are the *increments* of the state and control variables.

3. With this transformation, the difference of the state-space equation is:

$$
\begin{aligned}
\Delta x_m(k+1) &= A_m \Delta x_m(k) + B_m \Delta u(k) \\
y(k+1) &= y(k) + C_m \Delta x_m(k+1) \\
&= y(k) + C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k)
\end{aligned}
$$

4. Now we have an *augmented state-space model* as follow:

$$\begin{bmatrix}\Delta x_m(k+1)\\ y(k+1)\end{bmatrix} = \begin{bmatrix}A_m & 0_m^T\\ C_mA_m & 1\end{bmatrix}\begin{bmatrix}\Delta x_m(k)\\ y(k)\end{bmatrix} + \begin{bmatrix}B_m\\ C_mB_m\end{bmatrix}\Delta u(k) \quad \rightarrow \quad x(k+1) = Ax(k) + B\Delta u(k)$$

$$y(k) = \begin{bmatrix}0_m & 1\end{bmatrix}\begin{bmatrix}\Delta x_m(k)\\ y(k)\end{bmatrix} \quad \rightarrow \quad y(k) = Cx(k)$$

where $0_m = [0, 0, \cdots, 0]$ is n_1 dimensional zero row vector.

5. (Example 1.1) Consider a discrete-time model in the following form:

$$x_m(k+1) = A_mx_m(k) + B_mu(k) \qquad y(k) = C_mx_m(k)$$

$$A_m = \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix} \quad B_m = \begin{bmatrix}0.5\\ 1\end{bmatrix} \qquad C_m = \begin{bmatrix}1 & 0\end{bmatrix}$$

(Solution) Since $n_1 = 2$, $0_m = \begin{bmatrix}0 & 0\end{bmatrix}$. The *augmented model for this plant* is given by

$$x(k+1) = Ax(k) + B\Delta u(k) \qquad y(k) = Cx(k)$$

$$A = \begin{bmatrix}A_m & 0_m^T\\ C_mA_m & 1\end{bmatrix} = \begin{bmatrix}1 & 1 & 0\\ 0 & 1 & 0\\ 1 & 1 & 1\end{bmatrix} \quad B = \begin{bmatrix}B_m\\ C_mB_m\end{bmatrix} = \begin{bmatrix}0.5\\ 1\\ 0.5\end{bmatrix} \quad C = \begin{bmatrix}0_m & 1\end{bmatrix} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$$

The *characteristic equation* of matrix A is given by

$$\det(\lambda I - A) = \det\begin{bmatrix}\lambda I - A_m & 0_m^T\\ -C_mA_m & (\lambda-1)\end{bmatrix} = (\lambda-1)\det(\lambda I - A_m) = (\lambda-1)^3$$

Two eigenvalues are from the original integrator plant, and one is from the augmentation of the plant model.

6. (Matlab "extmodel.m") Consider the continuous-time system as follow:

$$\dot{x}_m(t) = A_m x_m(t) + B_m u(t) \qquad y(t) = C_m x_m(t)$$

$$A_m = \begin{bmatrix} 0 & 1 & 0 \\ 3 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad B_m = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \qquad C_m = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

where the sampling time $\Delta T = 1[s]$.

```
Ac = [0 1 0; 3 0 1;0 1 0];
Bc = [1; 1; 3];
Cc = [0 1 0];
Dc = zeros(1,1);
Delta_t = 1;
[Ad,Bd,Cd,Dd] = c2dm(Ac,Bc,Cc,Dc,Delta_t);

[m1,n1] = size(Cd);
[n1,n_in] = size(Bd);
A_e = eye(n1+m1,n1+m1);
A_e(1:n1,1:n1) = Ad;
A_e(n1+1:n1+m1,1:n1) = Cd * Ad;
B_e = zeros(n1+m1,n_in);
B_e(1:n1,:) = Bd;
B_e(n1+1:n1+m1,:) = Cd * Bd;
C_e = zeros(m1,n1+m1);
C_e(:,n1+1:n1+m1) = eye(m1,m1);
```

(MPC) 1.3 Predictive Control within One Optimization Window

1. Upon formulation of the mathematical model, the next step in the design of a predictive control system is to calculate the *predicted plant output with the future control signal* as the adjustable variables.

2. Assume that the current time is k and the length of the optimization window is N_p, as the number of samples.

3. Prediction of State and Output Variables

   a) Assuming that, at the sampling instant k, the state variable vector $x(k)$ is available through measurement, the state $x(k)$ provides the current plant information.

   b) The *future control trajectory* is denoted by

   $$\Delta u(k),\ \Delta u(k+1),\ \Delta u(k+2),\ \cdots, \Delta u(k+N_c-1)$$

   where N_c is called the *control horizon* dictating the number of parameters used to capture the future control trajectory.

   c) With given information $x(k)$, the future state variables are predicted for N_p, number of samples, where N_p is called the *prediction horizon*. N_p is also the length of the optimization window.

   d) The *future (or predicted) state variables* are denoted by

   $$x(k+1|k),\ x(k+2|k),\ x(k+3|k),\ \cdots, x(k+N_p|k)$$

where $x(k+m|k)$ is the predicted state variable at $k+m$ with given current plant information $x(k)$.

e) The control horizon N_c is chosen to be less than (or equal to) the prediction horizon N_p, namely $N_c \leq N_p$.

f) The *future state variables* are calculated sequentially using the set of future control parameters

$$
\begin{aligned}
x(k+1|k) &= Ax(k) + B\Delta u(k) \\
x(k+2|k) &= Ax(k+1|k) + B\Delta u(k+1) \\
&= A^2x(k) + AB\Delta u(k) + B\Delta u(k+1) \\
x(k+3|k) &= Ax(k+2|k) + B\Delta u(k+2) \\
&= A^3x(k) + A^2B\Delta u(k) + AB\Delta u(k+1) + B\Delta u(k+2) \\
&\vdots \\
x(k+N_p|k) &= A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + A^{N_p-2}B\Delta u(k+1) + \cdots + A^{N_p-N_c}B\Delta u(k+N_c-1)
\end{aligned}
$$

g) From the predicted state variables, the *predicted output variables* are, by substitution

$$\begin{aligned}
y(k+1|k) &= Cx(k+1|k) = CAx(k) + CB\Delta u(k) \\
y(k+2|k) &= Cx(k+2|k) = CA^2x(k) + CAB\Delta u(k) + CB\Delta u(k+1) \\
y(k+3|k) &= Cx(k+3|k) = CA^3x(k) + CA^2B\Delta u(k) + CAB\Delta u(k+1) + CB\Delta u(k+2) \\
&\vdots \\
y(k+N_p|k) &= CA^{N_p}x(k) + CA^{N_p-1}B\Delta u(k) + CA^{N_p-2}B\Delta u(k+1) + \cdots + CA^{N_p-N_c}B\Delta u(k+N_c-1)
\end{aligned}$$

h) Note that all predicted variables are formulated in terms of current state $x(k)$ and the future control movement $\Delta u(k+j)$, for $j = 0, 1, 2, \cdots, N_c - 1$. As a compact form,

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

$$\begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ y(k+3|k) \\ \vdots \\ y(k+N_p|k) \end{bmatrix} = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix} x(k) + \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-2}B & \cdots & CA^{N_p-N_c}B \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix}$$

i) This compact form will be utilized for the implementation of the MPC.

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

where $Y \in \Re^{N_p}$, $F \in \Re^{N_p \times n}$, $x(k) \in \Re^n$, $\Phi \in \Re^{N_p \times N_c}$, and $\Delta U \in \Re^{N_c}$

4. Optimization

a) For a given set-point signal $r(k) \in \Re$ at sample time k, within a prediction horizon, the objective of the predictive control system is to bring the predicted output as close as possible to the set-point signal.

$$\begin{aligned} R_s^T &= \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} r(k) \\ &= \bar{R}_s^T r(k) \end{aligned}$$

where $\bar{R}_s = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$ is a N_p-dimensional one column vector. On the other hand, for the trajectory tracking signal,

$$R_s^T = \begin{bmatrix} r(k+1) & r(k+2) & \cdots & r(k+N_p) \end{bmatrix}$$

where $R_s \in \Re^{N_p}$ has a future reference trajectory to be followed.

b) This objective is then translated into a design to find the *best* control parameter vector ΔU such that an error function between the set-point (or future reference) and the predicted output is minimized. Let us define the cost function J that reflects the control objective

$$J = \frac{1}{2}(R_s - Y)^T(R_s - Y) + \frac{1}{2}\Delta U^T \bar{R} \Delta U$$

where the control input weighting $\bar{R} = r_w I_{N_c \times N_c}$ is a diagonal matrix and r_w is a *tuning* parameter.

- when $r_w = 0$, we would not want to pay any attention to how large the ΔU might be.
- when $r_w \gg 0$, the cost function is interpreted as the situation where we would carefully consider how large the ΔU might be and cautiously reduce the error $|R_s - Y|$.

c) To find the optimal ΔU that will minimize J,

$$J = \frac{1}{2}(R_s - Fx(k) - \Phi\Delta U)^T(R_s - Fx(k) - \Phi\Delta U) + \frac{1}{2}\Delta U^T \bar{R}\Delta U$$
$$= \frac{1}{2}(R_s - Fx(k))^T(R_s - Fx(k)) - \Delta U^T\Phi^T(R_s - Fx(k)) + \frac{1}{2}\Delta U^T\Phi^T\Phi\Delta U + \frac{1}{2}\Delta U^T\bar{R}\Delta U$$

d) The *necessary condition* of the minimum J is obtained as

$$\frac{\partial J}{\partial \Delta U} = -\Phi^T(R_s - Fx(k)) + \Phi^T\Phi\Delta U + \bar{R}\Delta U = 0 \quad \rightarrow \quad \Delta U = (\Phi^T\Phi + \bar{R})^{-1}\Phi^T(R_s - Fx(k))$$

where the matrix $(\Phi^T\Phi + \bar{R})$ is called the *Hessian* matrix in the optimization literature.

e) In the set-point control case, note that $R_s = \bar{R}_s r(k)$. The optimal solution of the control signal is linked to the set-point signal $r(k)$ and the state variable $x(k)$:

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

5. (Matlab "mpcgain.m")

```
function [Phi_Phi, Phi_F, Phi_R, F, BarRs, Phi, A_e, B_e,C_e]
                                        = mpcgain(Ap, Bp, Cp, Nc, Np);
[m1,n1] = size(Cp);
[n1,n_in] = size(Bp);
A_e = eye(n1+m1,n1+m1);
A_e(1:n1,1:n1) = Ap;
A_e(n1+1:n1+m1,1:n1) = Cp * Ap;
B_e = zeros(n1+m1,n_in);
B_e(1:n1,:) = Bp;
```

```
B_e(n1+1:n1+m1,:) = Cp * Bp;
C_e = zeros(m1,n1+m1);
C_e(:,n1+1:n1+m1) = eye(m1,m1);

n = n1 + m1;
h(1,:) = C_e;
F(1,:) = C_e * A_e;

for kk=2:Np
     h(kk,:) = h(kk-1,:) * A_e;
     F(kk,:) = F(kk-1,:) * A_e;
end

v = h * B_e;
Phi = zeros(Np,Nc); %declare the dimension of Phi
Phi(:,1) = v; % first column of Phi

for i=2:Nc
     Phi(:,i) = [zeros(i-1,1); v(1:Np-i+1,1)]; %Toeplitz matrix
end

BarRs = ones(Np,1);
Phi_Phi = Phi' * Phi;
Phi_F = Phi' * F;
Phi_R = Phi' * BarRs;
```

6. (Example 1.2) Consider a first-order system

$$x_m(k+1) = 0.8x_m(k) + 0.1u(k) \qquad\qquad y(k) = x_m(k)$$

(1) Find the augmented state-space model?
(2) Calculate the components that form the prediction of future output Y, and the quantities $\Phi^T\Phi$, $\Phi^T F$, and $\Phi^T \bar{R}_s$ with $N_p = 10$ and $N_c = 4$?
(3) Assuming that, at a time $k = 10$, $r(k) = 1$ and the state vector $x(k) = [0.1\ ,\ 0.2]^T$, find the optimal solution ΔU with respect to the cases where $r_w = 0$ and $r_w = 10$, and compare the results?
(solution)
(1) The augmented state-space equation is

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} 0.8 & 0 \\ 0.8 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} \Delta u(k)$$

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

(2) Run “mpcgain.m”
(3) when $r_w = 0$, $r(k) = 1$ and $x(k) = [0.1, 0.2]^T$, Type

```
DelU = inv(Phi_Phi)*(Phi_R*1 - Phi'*F*[0.1 ; 0.2])
```

when $r_w = 10$, $r(k) = 1$ and $x(k) = [0.1, 0.2]^T$,, Type

```
DelU = inv(Phi_Phi + 10*eye(Nc,Nc))*(Phi_R*1 - Phi'*F*[0.1 ; 0.2])
```

7. (Example 1.3) Optimality can be proven using the completion of squares. Try it!

(MPC) 1.4 Receding Horizon Control

1. Among the controls $\Delta u(k), \Delta u(k+1), \cdots, \Delta u(k+N_c-1)$, the receding horizon control principle requires the *first sample* of this sequence, *i.e.,* $\Delta u(k)$ while ignoring the rest of the sequence.

2. When the next sample period arrives, the more recent measurement is taken to form the state vector $x(k+1)$ for calculation of the new sequence of control signal. This procedure is repeated in real time to give the receding horizon control law.

3. (Example 1.4) Consider a first-order system

$$x_m(k+1) = 0.8x_m(k) + 0.1u(k) \qquad\qquad y(k) = x_m(k)$$

where $N_p = 10$, $N_c = 4$, $r_w = 0$, $r(k) = 1$ for all k, at an initial time $k = 10$, the state vector $x(10) = [0.1\ ,\ 0.2]^T$ and $u(9) = 0$.
(solution)
At sample time $k = 10$,

$$\Delta U = (\Phi^T\Phi)^{-1}\Phi^T(\bar{R}_s r(k) - Fx(k)) = \begin{bmatrix}7.2 & -6.4 & 0 & 0\end{bmatrix}^T$$

$$u(10) = u(9) + \Delta u(10) = 0 + 7.2 = 7.2 \qquad x_m(10) = y(10) = 0.2$$

$$x_m(11) = 0.8x_m(10) + 0.1u(10) = 0.88 \qquad x(11) = \begin{bmatrix}\Delta x_m(11) \\ y(11)\end{bmatrix} = \begin{bmatrix}0.88 - 0.2 \\ 0.88\end{bmatrix} = \begin{bmatrix}0.68 \\ 0.88\end{bmatrix}$$

At sample time $k = 11$,

$$\Delta U = \begin{bmatrix}-4.24 & -0.96 & 0 & 0\end{bmatrix}^T$$

$$u(11) = u(10) + \Delta u(11) = 7.2 - 4.24 = 2.96$$

$$x_m(12) = 0.8x_m(11) + 0.1u(11) = 1 \qquad x(12) = \begin{bmatrix} \Delta x_m(12) \\ y(12) \end{bmatrix} = \begin{bmatrix} 0.12 \\ 1 \end{bmatrix}$$

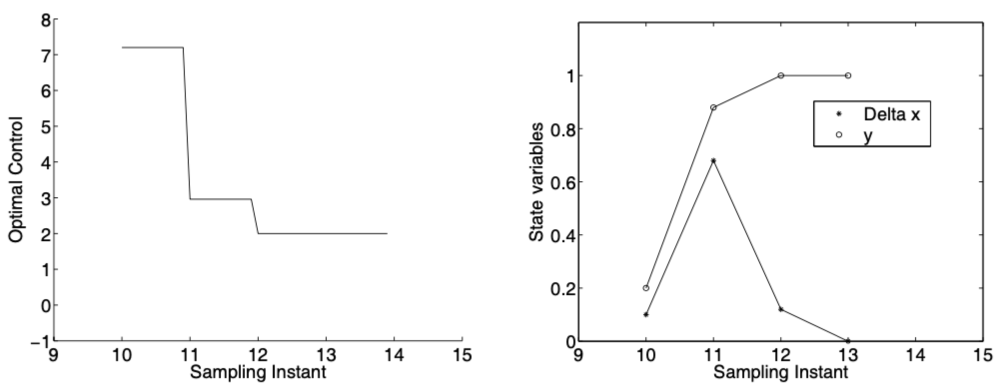
At sample time $k = 12$,

$$\Delta U = \begin{bmatrix} -0.96 & 0 & 0 & 0 \end{bmatrix}^T$$

$$u(12) = u(11) + \Delta u(12) = 2.96 - 0.96 = 2$$

$$x_m(13) = 0.8x_m(12) + 0.1u(12) = 1 \qquad x(13) = \begin{bmatrix} \Delta x_m(13) \\ y(13) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

At sample time $k = 13$, $\Delta U = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$



(a) Optimal control (b) State variable

Fig. 1.2. Receding horizon control

The figure shows the trajectories of the state variable $\Delta x_m(k)$ and $y(k)$, as well as the control signal that was used to regulate the output.

4. Closed-loop System by Set-point Control

a) Reconsider the optimal parameter vector at a given time k

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}\Phi^T\bar{R}_s r(k) - (\Phi^T\Phi + \bar{R})^{-1}\Phi^T F x(k)$$

where

$$\begin{aligned}
(\Phi^T\Phi + \bar{R})^{-1}\Phi^T\bar{R}_s r(k): &\quad \text{the set-point change} \\
(\Phi^T\Phi + \bar{R})^{-1}\Phi^T F: &\quad \text{state feedback control within the framework of predictive control}
\end{aligned}$$

b) Because of the receding horizon control principle, we only take the *first element* of ΔU at time k as the *incremental control*, thus

$$\begin{aligned}
\Delta u(k) &= [1\ 0\ \cdots\ 0](\Phi^T\Phi + \bar{R})^{-1}\Phi^T\bar{R}_s r(k) - [1\ 0\ \cdots\ 0](\Phi^T\Phi + \bar{R})^{-1}\Phi^T F x(k) \\
&= K_y r(k) - K_{mpc} x(k)
\end{aligned}$$

where

$$\begin{aligned}
K_y \in \Re: &\quad \text{first element of } (\Phi^T\Phi + \bar{R})^{-1}\Phi^T\bar{R}_s \\
K_{mpc} \in \Re^{1\times n}: &\quad \text{first row of } (\Phi^T\Phi + \bar{R})^{-1}\Phi^T F
\end{aligned}$$

c) Let us apply above incremental control to the augmented system

$$\begin{aligned}
x(k+1) &= Ax(k) + B\Delta u(k) \\
&= Ax(k) + BK_y r(k) - BK_{mpc} x(k) \\
&= [A - BK_{mpc}]x(k) + BK_y r(k)
\end{aligned}$$

d) *Characteristic equation*:

$$\det[\lambda I - (A - BK_{mpc})] = 0$$

e) Because of the special structures of the matrices C and A, the *last column* of F is identical to $\bar{R}_s$, which is $[1\ 1\ \cdots\ 1]$, therefore K_y is identical to the *last element* of K_{mpc}.

$$K_{mpc} = [K_x\ K_y]$$

where $K_x \in \Re^{1\times n_1}$ and $K_y \in \Re$.

f) (Example 1.5) Reconsider the first-order system

$$x_m(k+1) = 0.8x_m(k) + 0.1u(k) \qquad y(k) = x_m(k)$$

where $N_p = 10$, $N_c = 4$, $r(k) = 1$ for all k, find the closed-loop feedback matrices when $r_w = 0$ and $r_w = 10$?
(Solution)
When $r_w = 0$, we have

$$K_y = [1\ 0\ 0\ 0](\Phi^T\Phi + r_w I_{4\times4})^{-1}\Phi^T \begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}^T = 10$$
$$k_{mpc} = [1\ 0\ 0\ 0](\Phi^T\Phi + r_w I_{4\times4})^{-1}\Phi^T F = [8\ 10]$$

When $r_w = 10$, we have

$$K_y = [1\ 0\ 0\ 0](\Phi^T\Phi + r_w I_{4\times4})^{-1}\Phi^T \begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}^T = 0.2453$$
$$k_{mpc} = [1\ 0\ 0\ 0](\Phi^T\Phi + r_w I_{4\times4})^{-1}\Phi^T F = [0.6939\ 0.2453]$$

5. (Matlab "reced_2nd.m") for (Example 1.6) Suppose that a continuous-time system is described by the transfer function

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

where $\omega_n = 10$, $\zeta = 0.5$, $\Delta t = 0.01[s]$, $N_c = 3$, $N_p = 20$, $\bar{R} = r_w I_{N_c \times N_c}$, and $r_w = 0.5$. Obtain the step response ?

```
omega = 10;
zeta = 0.5;
numc = omega^2;
denc = [1 2*zeta*omega omega^2];
[Ac,Bc,Cc,Dc] = tf2ss(numc,denc);

Delta_t = 0.01;
[Ap,Bp,Cp,Dp] = c2dm(Ac,Bc,Cc,Dc,Delta_t);

Nc = 3;
Np = 20;
rw = 0.5;
[Phi_Phi, Phi_F, Phi_R, F, BarRs, Phi, A_e, B_e, C_e]
                  = mpcgain(Ap, Bp, Cp, Nc, Np);

[n,n_in] = size(B_e);
xm = [0;0];
Xf = zeros(n,1);
```

```
N_sim=100;
r = ones(N_sim,1);
u=0; % u(k-1) = 0
y=0;

for kk=1:N_sim;
    DeltaU = inv(Phi_Phi+rw*eye(Nc,Nc))*(Phi_R*r(kk) -Phi_F*Xf);
    deltau = DeltaU(1,1);
    u = u + deltau;
    u1(kk) = u;
    y1(kk) = y;
    xm_old = xm;
    xm = Ap * xm + Bp * u;
    y = Cp * xm ;
    Xf = [xm-xm_old;y];
end
k = 0:(N_sim-1);
figure
subplot(211)
plot(k,y1)
xlabel('Sampling Instant')
legend('Output')
subplot(212)
plot(k,u1)
xlabel('Sampling Instant')
legend('Control')
```

(MPC) 1.5 Predictive Control of MIMO Systems

1. For MIMO systems, assume that the plant has m *inputs,* q *outputs and* n_1 *states*.

2. In the general formulation of the predictive control problem, we also take the plant noise and disturbance into consideration.

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) + B_d w(k) \\ y(k) &= C_m x_m(k) \end{aligned}$$

   where $w(k)$ is the *input disturbance*, assumed to be a sequence of *integrated white noise*. This means that the input disturbance $w(k)$ is related to a *zero-mean, white noise* sequence $\epsilon(k)$ by the difference equation

$$w(k) - w(k-1) = \epsilon(k)$$

3. By defining $\Delta x_m(k) = x_m(k) - x_m(k-1)$ and $\Delta u(k) = u(k) - u(k-1)$, we have

$$\begin{aligned} \Delta x_m(k+1) &= A_m \Delta x_m(k) + B_m \Delta u(k) + B_d \epsilon(k) \\ y(k+1) - y(k) &= C_m \Delta x_m(k+1) = C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) + C_m B_d \epsilon(k) \end{aligned}$$

4. Choosing *new state* variable vector $x(k) = [\Delta x_m(k)^T \; y(k)^T]^T \in \Re^{n_1+q}$, we have

$$\begin{aligned} \begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} &= \begin{bmatrix} A_m & 0_{n_1\times q} \\ C_m A_m & I_{q\times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) + \begin{bmatrix} B_d \\ C_m B_d \end{bmatrix} \epsilon(k) \\ y(k) &= \begin{bmatrix} 0_{q\times n_1} & I_{q\times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \end{aligned}$$

5. For notational simplicity, we introduce the *augmented state-space model* as follow:

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) + B_d\epsilon(k) \\ y(k) &= Cx(k) \end{aligned}$$

   where $x(k) \in \Re^n$ with $n = n_1 + q$, $A \in \Re^{n\times n}$, $B \in \Re^{n\times m}$, and $C \in \Re^{q\times n}$

6. Eigenvalues of the augmented model are obtained by characteristic polynomial equation

$$\begin{aligned} \det[\lambda I - A] &= \det\begin{bmatrix} \lambda I_{n_1\times n_1} - A_m & 0_{n_1\times q} \\ -C_m A_m & (\lambda-1)I_{q\times q} \end{bmatrix} \\ &= (\lambda-1)^q \det[\lambda I_{n_1\times n_1} - A_m] = 0 \end{aligned}$$

   where the eigenvalues of the augmented model are the union of the eigenvalues of the plant model and the q eigenvalues, $\lambda = 1$.

7. This means that there are q integrators embedded into the augmented design model. This is the means we use to obtain integral action for the MPC systems.

8. Stabilizability (Controllability) / Detectability (Observability)

9. Minimal Realization (no pole-zero cancelation) guarantees controllability and observability of the control system. For example

$$G(z) = \frac{(z-0.1)}{(z-0.1)(z-0.9)} \quad \textbf{non-minimal} \quad A_m = \begin{bmatrix} 1 & -0.09 \\ 1 & 0 \end{bmatrix} \quad B_m = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad C_m = \begin{bmatrix} 1 & -0.1 \end{bmatrix}$$

For minimal realization, matlab code is

```
numd = [1 -0.1];
dend = conv([1 -0.1],[1 -0.9]);
sys1 = tf(numd,dend) ;
sys = ss(sys1,'min');
[Am,Bm,Cm,Dm] = ssdata(sys)
```

The minimal realization through model-order reduction is

$$A_m = 0.9 \quad B_m = -0.9285 \quad C_m = -1.077 \qquad \textbf{minimal} \qquad G(z) = \frac{1}{z-0.9}$$

10. Solution of Predictive Control for MIMO Systems

Define the vectors Y and ΔU as

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

Based on the state-space model (A, B, C), the *future state* variables are calculated sequentially using the set of *future control* parameters

$$\begin{aligned} x(k+1|k) &= Ax(k) + B\Delta u(k) + B_d\epsilon(k) \\ x(k+2|k) &= Ax(k+1|k) + B\Delta u(k+1) + B_d\epsilon(k+1|k) \\ &= A^2x(k) + AB\Delta u(k) + B\Delta u(k+1) + AB_d\epsilon(k) + B_d\epsilon(k+1|k) \\ &\vdots \\ x(k+N_p|k) &= A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + A^{N_p-2}B\Delta u(k+1) + \cdots + A^{N_p-N_c}B\Delta u(k+N_c-1) \\ &\quad + A^{N_p-1}B_d\epsilon(k) + A^{N_p-2}B_d\epsilon(k+1|k) + \cdots + B_d\epsilon(k+N_p-1|k) \end{aligned}$$

With the assumption that $\epsilon(k)$ is a *zero-mean white noise* sequence, the *predicted value* of of $\epsilon(k+i|k)$ at future sample i is assumed to be *zero*. The prediction of the state variable and *output variable* is calculated as the expected values being zero. Effectively, we have

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

$$\begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ y(k+3|k) \\ \vdots \\ y(k+N_p|k) \end{bmatrix} = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix} x(k) + \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-2}B & \cdots & CA^{N_p-N_c}B \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix}$$

The *incremental optimal control* within one optimization window is given by

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

where $\Phi^T\Phi \in \Re^{mN_c \times mN_c}$, $\Phi^T F \in \Re^{mN_c \times n}$, $\Phi^T \bar{R}_s$ equals the last q columns of $\Phi^T F$. Applying the *receding horizon control principle*, the first m elements in ΔU are taken to form the incremental optimal control:

$$\begin{aligned}\Delta u(k) &= \begin{bmatrix} I_{m\times m} & 0_{m\times m} & 0_{m\times m} & \cdots & 0_{m\times m}\end{bmatrix} (\Phi^T\Phi + \bar{R})^{-1}\Phi^T(\bar{R}_s r(k) - Fx(k)) \\ &= K_y r(k) - K_{mpc}x(k)\end{aligned}$$

(MPC) 1.6 State Estimation

1. Till now, we assumed that all the state variables are measurable or available, but some of them may be impossible to measure. Thus we need *observer* to provide the state estimates.

2. Our focus here is to use an observer in the design of predictive control.

3. Basic Ideas About an Observer (Luenberger Observer)

   a) For given plant state model,

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

   the typical Luenberger observer is designed as following form:

   $$\hat{x}_m(k+1) = A_m \hat{x}_m(k) + B_m u(k) + K_{ob}(y(k) - C_m \hat{x}_m(k))$$

   where K_{ob} is the *observer gain* matrix.

   b) To choose the observer gain K_{ob}, we examine the closed-loop error dynamics with error state $\tilde{x}_m(k) = x_m(k) - \hat{x}_m(k)$

   $$\begin{aligned}\tilde{x}_m(k+1) &= A_m \tilde{x}_m(k) - K_{ob}(y(k) - C_m \hat{x}_m(k)) \\ &= (A_m - K_{ob} C_m)\tilde{x}_m(k)\end{aligned}$$

   Now, with given initial error $\tilde{x}_m(0)$, we have

   $$\tilde{x}_m(k) = (A_m - K_{ob} C_m)^k \tilde{x}_m(0)$$

   where the observer gain can be used to manipulate the convergence rate of the error.

c) (Example 1.7) Consider the linearized pendulum equation

$$\ddot{\theta} + \omega_n^2 \theta = u$$

Design an observer that reconstructs the angle θ of the pendulum given measurements of $\dot{\theta}$, namely $y = \dot{\theta}$, where $\omega_n = 2$, $\Delta t = 0.1[s]$, and the desired observer poles are chosen to be 0.1 and 0.2 ?

(Solution)

Let $x_1 = \theta$ and $x_2 = \dot{\theta}$, the model is obtained by

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

The corresponding discrete-time model is obtained using the matlab function $c2dm(A, B, C, D, \Delta t)$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9801 & 0.0993 \\ -0.3973 & 0.9801 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.0050 \\ 0.0993 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

Assume that the observer gain $K_{ob} = [j_1, j_2]^T \in \Re^2$. The closed-loop characteristic polynomial for the observer is

$$\det(\lambda I - A_m + K_{ob} C_m) = \det\left(\lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.9801 & 0.0993 \\ -0.3973 & 0.9801 \end{bmatrix} + \begin{bmatrix} j_1 \\ j_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix}\right)$$

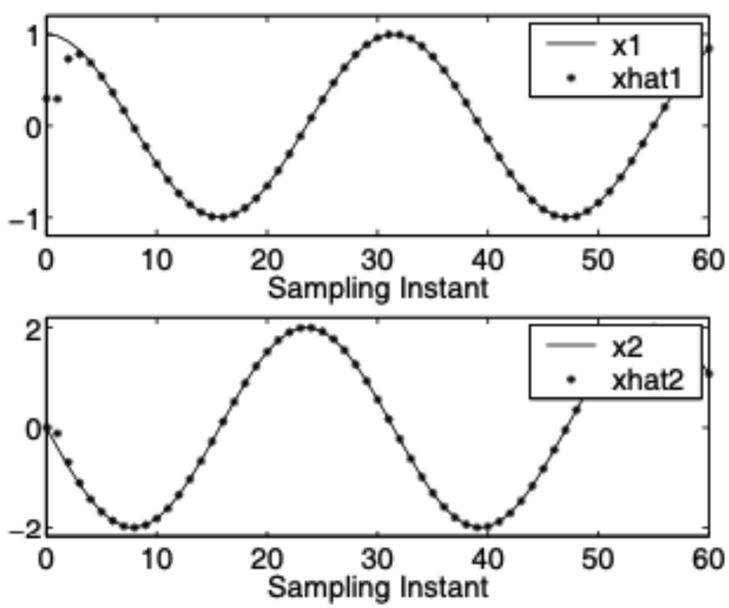
$$= \det \begin{bmatrix} \lambda - 0.9801 & -0.0993 + j_1 \\ 0.3973 & \lambda - 0.9801 + j_2 \end{bmatrix} = (\lambda - 0.1)(\lambda - 0.2)$$

Solution of polynomial equation gives us the observer gain as

$$\therefore \quad j_1 = -1.6284 \qquad j_2 = 1.6601$$

Now we have finished the observer design

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9801 & 0.0993 \\ -0.3973 & 0.9801 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + \begin{bmatrix} 0.0050 \\ 0.09930 \end{bmatrix} u(k) + \begin{bmatrix} -1.6284 \\ 1.6601 \end{bmatrix} (x_2(k) - \hat{x}_2(k))$$



(b) Estimation with observer

When $u(k) = 0$, $x_1(0) = 1$, $x_2(0) = 0$, $\hat{x}_1(0) = 0.3$, and $\hat{x}_2(0) = 0$

(MPC) 1.7 State Estimate Predictive Control

1. In the implementation of predictive control, an observer is used for the cases where the state variable $x(k)$ at time k is not measurable. Essentially, the state variable $x(k)$ is estimated via an *observer* of the form:

$$\hat{x}(k+1) = A\hat{x}(k) + B\Delta u(k) + K_{ob}(y(k) - C\hat{x}(k))$$

2. With the information of $\hat{x}(k)$ replacing $x(k)$, the predictive control law is then modified to find ΔU by minimizing

$$J = \frac{1}{2}(R_s - F\hat{x}(k))^T(R_s - F\hat{x}(k)) - \Delta U^T\Phi^T(R_s - F\hat{x}(k)) + \frac{1}{2}\Delta U^T\left(\Phi^T\Phi + \bar{R}\right)\Delta U$$

3. The optimal solution is obtained as

$$\frac{\partial J}{\partial \Delta U} = 0 \quad \rightarrow \quad \Delta U = (\Phi^T\Phi + \bar{R})^{-1}\Phi^T(R_s - F\hat{x}(k))$$

4. Application of the receding horizon control principle leads to the optimal solution of $\Delta u(k)$ at time k:

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

5. Standard state-space feedback control structure based on the estimated $\hat{x}(k)$ is illustrated in the following figure

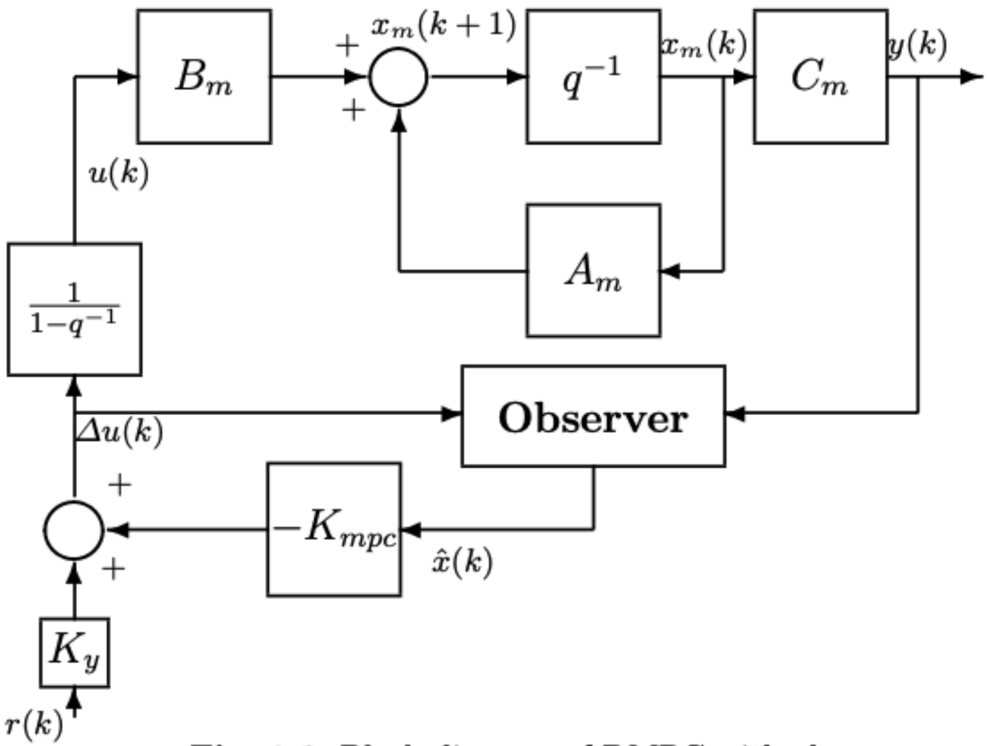


Fig. 1.6. Block diagram of DMPC with observer

6. Separation Principle (between control eigenvalues and observer eigenvalues)

a) Let us obtain the closed-loop control system using $\tilde{x} = x - \hat{x}$

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) = Ax(k) - BK_{mpc}\hat{x}(k) + BK_y r(k) \\ &= (A - BK_{mpc})x(k) - BK_{mpc}\tilde{x}(k) + BK_y r(k) \\ \tilde{x}(k+1) &= (A - K_{ob}C)\tilde{x}(k) \end{aligned}$$

b) Combining above both equations, we have

$$\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1) \end{bmatrix} = \begin{bmatrix} A - BK_{mpc} & -BK_{mpc} \\ 0_{n\times n} & A - K_{ob}C \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} BK_y \\ 0_{n\times m} \end{bmatrix} r(k)$$

c) *Characteristic equation* of the closed-loop control system is determined by

$$\det \left[\lambda \begin{bmatrix} I_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & I_{n\times n} \end{bmatrix} - \begin{bmatrix} A - BK_{mpc} & -BK_{mpc} \\ 0_{n\times n} & A - K_{ob}C \end{bmatrix} \right] = \det[\lambda I_{n\times n} - A + BK_{mpc}] \cdot \det[\lambda I_{n\times n} - A + K_{ob}C] = 0$$

d) The closed-loop model predictive control system with state estimate has two independent characteristic equations:

$$\begin{aligned} \det[\lambda I_{n\times n} - A + BK_{mpc}] &= 0 \\ \det[\lambda I_{n\times n} - A + K_{ob}C] &= 0 \end{aligned}$$

This means that the *design* of the predictive control law and the observer can be carried out *independently (or separately)*, since the eigenvalues remain unchanged.