# *Model Predictive Control Problem Formulation*

The objective of a model predictive control strategy is to:

*Compute a trajectory of future control inputs that optimizes the future behavior of plant output, where the optimization is carried out within a limited time window*

## An Application Example

- Let's examine how this might work for a golf game.
  - Prior to tee-off, we analyze the course and game-time weather conditions.
  - Based on this knowledge - and the physical principles governing ball trajectories as a function of club - we develop a *model.*
  - Using this model, we develop a strategy for the next $n$ golf strokes that will land the ball in the hole.
  - We execute the next shot in this optimal sequence.
  - We then assess our success and use information gained to plan the next series of shots.
- Of course a key difference in golf is that our horizon is ever-shortening (we hope!)
- In practice, our actual play sequence will differ from the original set of *n* shots in the optimal sequence.

- A part of this difference is due to unknowns (like how close to our potential we perform).

NOTE: In practice, we will have limitations (e.g., known injuries, maximum shot distance, etc.) that we will build in as *constraints* during pre-game analysis.

Key principles of MPC demonstrated by this example include:

1. Moving prediction horizon window
2. Receding horizon control that only executes the next step
3. Need for current-time information to make the next prediction
4. Model that describes system dynamics used for prediction
5. Objective criterion based on measured difference between desired & actual response

## Model Choice for MPC

- Three popular approaches to system modeling for predictive control design are:
  - Finite Impulse Response (FIR)
  - Step Response (SR)
  - Transfer Function (TF)
  - State-Space (SS)
- Each approach exhibits certain advantages and disadvantages - the development in this course will focus on *state-space models*
- FIR, SR and TF models are of historical interest and will be presented for completeness
- All models will assume a discrete-time setting

## *Finite Impulse Response*

- Also known as weighting sequence, the output is related to the input via the convolution operator
- For the single-input-single-output (SISO) case, we write

$$y_k = \sum_{i=1}^{N} g_i u_{k-i}$$

  where $u_k$ and $y_k$ denote the input and output sequence elements, respectively.
- The sequence $g_i$ comprises the first $N$ coefficients of an infinite Taylor series in the unit time delay operator, $z^{-1}$ , representing the dynamic process $g(z)$:

$$g(z) = \sum_{i=0}^{\infty} g_i z^{-1}$$

  – This sequence will always decay to zero for the stable case.
- The first $N$ values of the infinite series give a *finite* impulse response which will capture the main process dynamics.
- A FIR model can be viewed as a representation of the discrete-time *pulse response* (expected value of the output $y$ in response to a single unit pulse $u$).
- This model generalizes easily to the multi-input-multi-output (MIMO) case by use of appropriate matrices.

## *Step Response*

- Integrating the unit pulse response will give the *unit step response*,

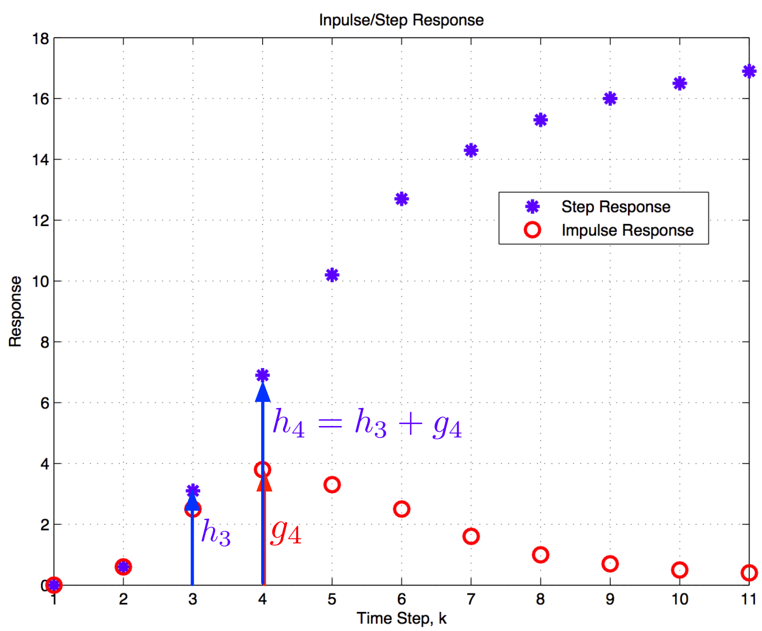
$$h(z) = \sum_{i=1}^{\infty} h_i z^{-1}$$

where

$$h(z) = \frac{g(z)}{\triangle(z)}$$

and $\triangle(z)$ is the one-step difference operator, $1 - z^{-1}$.

- The following plot shows the relationship between the impulse and step response.




## *Transfer Function*

- Again considering the SISO case we can write

$$a(z)y_k = b(z)u_k$$

where the form $\dfrac{b(z)}{a(z)}$ gives a transfer function representation for the output-input ratio of the corresponding z-transforms.

- The polynomials assume the forms:

$$a(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}$$
$$b(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}$$

NOTE: Performing the long division $b(z)/a(z)$ will yield the corresponding unit pulse response $g(z)$.

*State-Space*

- We shall assume a linearized, discrete-time, state-space model of the plant:

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k)$$
$$\boldsymbol{y}(k) = C_y \boldsymbol{x}(k)$$
$$\boldsymbol{z}(k) = C_z \boldsymbol{x}(k)$$

where $\boldsymbol{x}$ is an $n$-dimensional state vector, $\boldsymbol{u}$ is an $\ell$-dimensional input vector, $\boldsymbol{y}$ is an $m_y$-dimensional vector of measured outputs, and $\boldsymbol{z}$ is an $m_z$-dimensional vector of outputs which are to be controlled.

- Since there is always finite delay between measuring $\boldsymbol{y}(k)$ and applying $\boldsymbol{u}(k)$, there is no direct feed-through term.
  - Hence our model is *strictly proper*.
- For simplicity, we shall often assume $\boldsymbol{y} \equiv \boldsymbol{z}$ in which case $C$ will denote both $C_y$ and $C_z$.

*Relationship to FIR Model*

We can use the discrete-time state equations to determine the sequence parameters resulting from a unit pulse input:

$$x(0) = 0$$

$$
\begin{aligned}
x(1) &= B \\
x(2) &= AB \\
x(3) &= A^2B \\
&\vdots \\
x(n) &= A^{n-1}B
\end{aligned}
$$

$$
\begin{aligned}
y(0) &= Cx(0) + Du(0) \\
&= D \\
y(1) &= CB \\
y(2) &= CAB \\
y(3) &= CA^2B \\
&\vdots \\
y(n) &= CA^{n-1}B
\end{aligned}
$$

- Hence the FIR model sequence is given by:

$$g(z) = D + CBz^{-1} + CABz^{-2} + CA^2Bz^{-3} + CA^{n-1}Bz^{-n}$$

*Relationship to Transfer Function Model*

- Applying the z-transform to the state-space model,

$$
\begin{aligned}
zX(z) &= AX(z) + BU(z) \\
(zI - A)X(z) &= BU(z) \\
X(z) &= (zI - A)^{-1}BU(z)
\end{aligned}
$$

and

$$Y(z) = C(zI - A)^{-1}BU(z)$$

- So that,

$$G(z) = \frac{Y(z)}{U(z)} = C(zI - A)^{-1}B$$

*State-Space Model with Embedded Integrator*

- Returning to our state-space representation, we consider the SISO system

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

- Recall from your study of feedback control that in order to eliminate steady-state error from a control system we need to include integral action.
- For model predictive controllers, a convenient way to introduce integral action is by modifying the state-space model.
- Taking a difference operation on both sides of the state equation we get

$$\boldsymbol{x}_m(k+1) - \boldsymbol{x}_m(k) = A_m(\boldsymbol{x}_m(k) - \boldsymbol{x_m}(k-1)) + B_m(u(k) - u(k-1))$$

- Denote the difference of the state variable by

$$\begin{aligned} \triangle \boldsymbol{x}_m(k+1) &= \boldsymbol{x_m}(k+1) - \boldsymbol{x_m}(k) \\ \triangle \boldsymbol{x}_m(k) &= \boldsymbol{x_m}(k) - \boldsymbol{x_m}(k-1) \end{aligned}$$

and of the control variable by

$$\triangle u(k) = u(k) - u(k-1)$$

- Then the difference of the state equation is:

$$\triangle \boldsymbol{x_m}(k+1) = A_m \triangle \boldsymbol{x_m}(k) + B_m \triangle u(k)$$

- We now have to compute the corresponding expression for the output $y(k)$.
- Define a new augmented state vector

$$\boldsymbol{x}(k) = \begin{bmatrix} \triangle \boldsymbol{x_m}(k) \\ y(k) \end{bmatrix}$$

- Note that

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

- Putting the above expressions together we get the augmented model:

$$\begin{bmatrix} \triangle \boldsymbol{x_m}(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & 0_m^T \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} \triangle \boldsymbol{x_m}(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \triangle u(k)$$

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

or simply,

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B \triangle u(k)$$

$$y(k) = C \begin{bmatrix} \triangle \boldsymbol{x_m}(k) \\ y(k) \end{bmatrix}$$

**Example (Wang, pg. 34)**

- Given a discrete-time system where

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

- Find the matrices of the augmented system $(A, B, C)$ and calculate the eigenvalues of $A$.

SOLUTION.

- We have $n = 2$ and $\mathbf{0}_m = \begin{bmatrix} 0 & 0 \end{bmatrix}$.

- Then,

$$\begin{aligned} \boldsymbol{x}(k+1) &= A\boldsymbol{x}(k) + B\triangle u(k) \\ y(k) &= C\boldsymbol{x}(k) \end{aligned}$$

where

$$A = \begin{bmatrix} A_m & 0_m^T \\ C_m A_m & I \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix}$$

$$C = \begin{bmatrix} 0_m & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

- The eigenvalues of $A$ are found as the roots of the characteristic equation, $(\lambda - 1)^3 = 0$; or $\lambda_1 = \lambda_2 = \lambda_3 = 1$.

## *Predictive Control: SISO Case*

- We will first develop the case for a simple SISO system for which the states are directly measureable.
  - The more general case will be discussed later.
- Future control inputs are denoted by:

$$\triangle u(k_i),\ \triangle u(k_i+1),\ \ldots,\ \triangle u(k_i+N_c-1)$$

where $k_i$ represents a particular time step and $N_c$ is the *control horizon*.

- Information $x(k_i)$ permits prediction up to $N_p$ samples ahead; $N_p$ is called the *prediction horizon.*

- Future state variables are given by:

$$x(k_i+1|k_i),\ x(k_i+2|k_i),\ \ldots\ x(k_i+m|k_i),\ \ldots,\ x(k_i+N_p|k_i)$$

NOTE: $N_c \le N_p$.

PREDICTION.

- Future state variables:

$$\begin{aligned}
\boldsymbol{x}(k_i+1|k_i) &= A\boldsymbol{x}(k_i) + B\triangle u(k_i)\\
\boldsymbol{x}(k_i+2|k_i) &= A\boldsymbol{x}(k_i+1|k_i) + B\triangle u(k_i+1)\\
&= A^2\boldsymbol{x}(k_i) + AB\triangle u(k_i) + B\triangle u(k_i+1)\\
&\vdots\\
\boldsymbol{x}(k_i+N_p|k_i) &= A^{N_p}\boldsymbol{x}(k_i) + A^{N_p-1}B\triangle u(k_i) + A^{N_p-2}B\triangle u(k_i+1)\\
&\quad + \ldots + A^{N_p-N_c}B\triangle u(k_i+N_c-1)
\end{aligned}$$

- Predicted outputs:

$$\begin{aligned}
y(k_i+1) &= CA\boldsymbol{x}(k_i) + CB\triangle u(k_i)\\
y(k_i+2) &= CA^2\boldsymbol{x}(k_i) + CAB\triangle u(k_i) + CB\triangle u(k_i+1)\\
y(k_i+3) &= CA^3\boldsymbol{x}(k_i) + CA^2B\triangle u(k_i) + CAB\triangle u(k_i+1)\\
&\quad + CB\triangle u(k_i+2)\\
&\vdots\\
y(k_i+N_p|k_i) &= CA^{N_p}\boldsymbol{x}(k_i) + CA^{N_p-1}B\triangle u(k_i) + CA^{N_p-2}B\triangle u(k_i+1)\\
&\quad + \ldots + CA^{N_p-N_c}B\triangle u(k_i+N_c-1)
\end{aligned}$$

- Gathering up the the predicted output sequence into the vector

$$
\boldsymbol{Y} = \begin{bmatrix} y(k_i+1) \mid k_i \\ y(k_i+2) \mid k_i \\ y(k_i+3) \mid k_i \\ \vdots \\ y(k_i+N_p) \mid k_i \end{bmatrix}
$$

- Gather future input increments into the vector

$$
\boldsymbol{\Delta U} = \begin{bmatrix} \triangle u(k_i) \\ \triangle u(k_i+1) \\ \triangle u(k_i+2) \\ \vdots \\ \triangle u(k_i+N_c-1) \end{bmatrix};
$$

- Now we can write

$$
\boldsymbol{Y} = F\boldsymbol{x}(k_i) + \Phi\boldsymbol{\Delta U}
$$

where

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

OPTIMIZATION.

- For a given set-point signal $r(k_i)$ at sample time $k_i$, the objective is to find the best control parameter $\boldsymbol{\Delta U}$ such that an error function is minimized.
- Define

$$\boldsymbol{R}_s = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} r(k_i)$$

- Then we can define a cost function $J$ as

$$J = (\boldsymbol{R_s} - \boldsymbol{Y})^t(\boldsymbol{R_s} - \boldsymbol{Y}) + \boldsymbol{\Delta U}^{\mathrm{t}} R \boldsymbol{\Delta U}$$

- Optimal $\boldsymbol{\Delta U}$ that minimizes $J$ is found from the solution of

$$\frac{\partial J}{\partial \boldsymbol{\Delta U}} = 0$$

- Expand $J$ as

$$J = (R_s - F\boldsymbol{x}(k_i))^t(R_s - F\boldsymbol{x}(k_i)) - 2\boldsymbol{\Delta U^t}\Phi^T(R_s - F\boldsymbol{x}(k_i)) + \boldsymbol{\Delta U^t}(\Phi^T\Phi + R)\boldsymbol{\Delta U}$$

to give

$$\frac{\partial J}{\partial \boldsymbol{\Delta U}} = -2\Phi^T(R_s - F\boldsymbol{x}(k_i)) + 2(\Phi^T\Phi + R)\boldsymbol{\Delta U} = 0$$

- Solving,

$$\boldsymbol{\Delta U^*} = (\Phi^T\Phi + R)^{-1}\Phi^T(R_s - F\boldsymbol{x}(k_i))$$

## Example

- Consider a simple first-order system:

$$x_m(k+1) = ax_m(k) + bu(k)$$
$$y(k) = cx_m(k)$$

where, $a = 0.8,\ b = 0.1 \frown\lll\Cup\ c = 1.0$. Let $N_p = 10$ and $N_c = 4$.

- Assume $k_i = 10$, and at this value, the reference $r(k_i) = 1$ and the state vector $\boldsymbol{x}(k_i) = \begin{bmatrix} 0.1 & 0.2 \end{bmatrix}^t$.

- Determine the optimal solution $\triangle U^*$ for the cases: 1) $R = [0]$ and 2) $R = 10 \cdot I$, where $R$ is the weighting on the control input.

SOLUTION.

- Form the augmented state-space system:

$$\begin{bmatrix} \boldsymbol{\Delta x_m}(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} a & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta x_m}(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} b \\ b \end{bmatrix} \triangle u(k)$$

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

- Therefore, $F$ and $\Phi$ take the following form:

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \\ CA^6 \\ CA^7 \\ CA^8 \\ CA^9 \\ CA^{10} \end{bmatrix}; \quad \Phi = \begin{bmatrix} CB & 0 & 0 & 0 \\ CAB & CB & 0 & 0 \\ CA^2B & CAB & CB & 0 \\ CA^3B & CA^2B & CAB & CB \\ CA^4B & CA^3B & CA^2B & CAB \\ CA^5B & CA^4B & CA^3B & CA^2B \\ CA^6B & CA^5B & CA^4B & CA^3B \\ CA^7B & CA^6B & CA^5B & CA^4B \\ CA^8B & CA^7B & CA^6B & CA^5B \\ CA^9B & CA^8B & CA^7B & CA^6B \end{bmatrix}$$

- Note that the elements of $\Phi$ comprise the discrete pulse response of the system arranged in the form of a Toeplitz matrix.

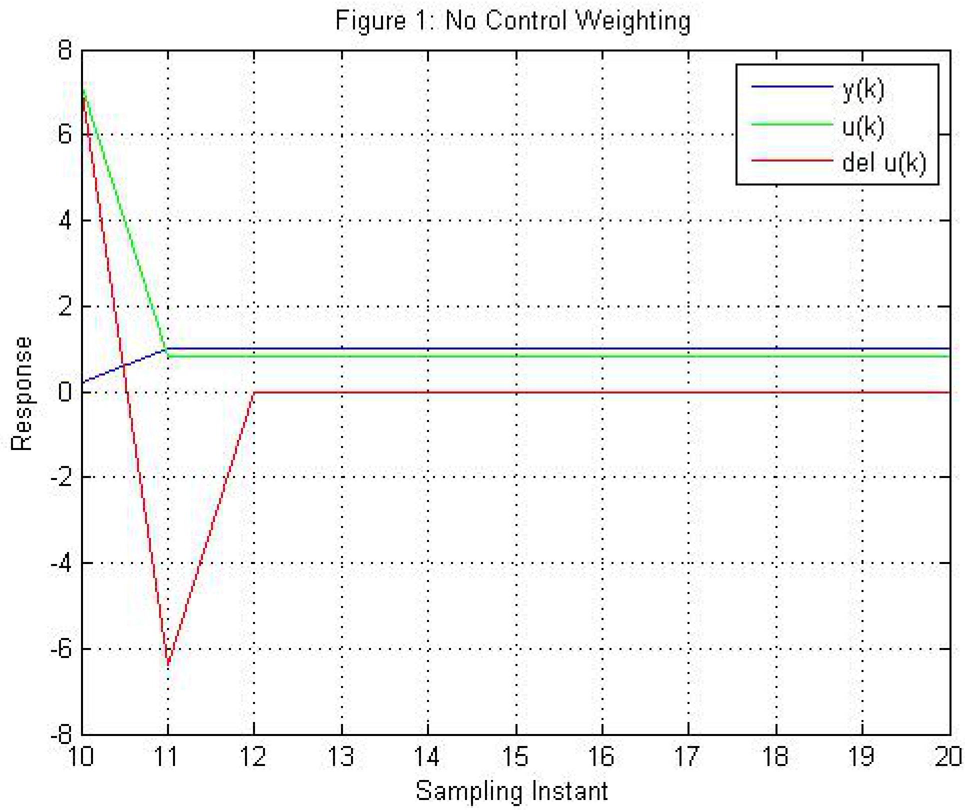
- Substituting values we obtain:

$$\Phi^T\Phi = \begin{bmatrix} 1.1541 & 1.0407 & 0.9116 & 0.7726 \\ 1.0407 & 0.9549 & 0.8475 & 0.7259 \\ 0.9116 & 0.8475 & 0.7675 & 0.6674 \\ 0.7726 & 0.7259 & 0.6674 & 0.5943 \end{bmatrix}$$

$$\Phi^T F = \begin{bmatrix} 9.2325 & 3.2147 \\ 8.3259 & 2.7684 \\ 7.2927 & 2.3355 \\ 6.1811 & 1.9194 \end{bmatrix}; \quad \Phi^T R_s = \begin{bmatrix} 3.2147 \\ 2.7684 \\ 2.3355 \\ 1.9194 \end{bmatrix}$$

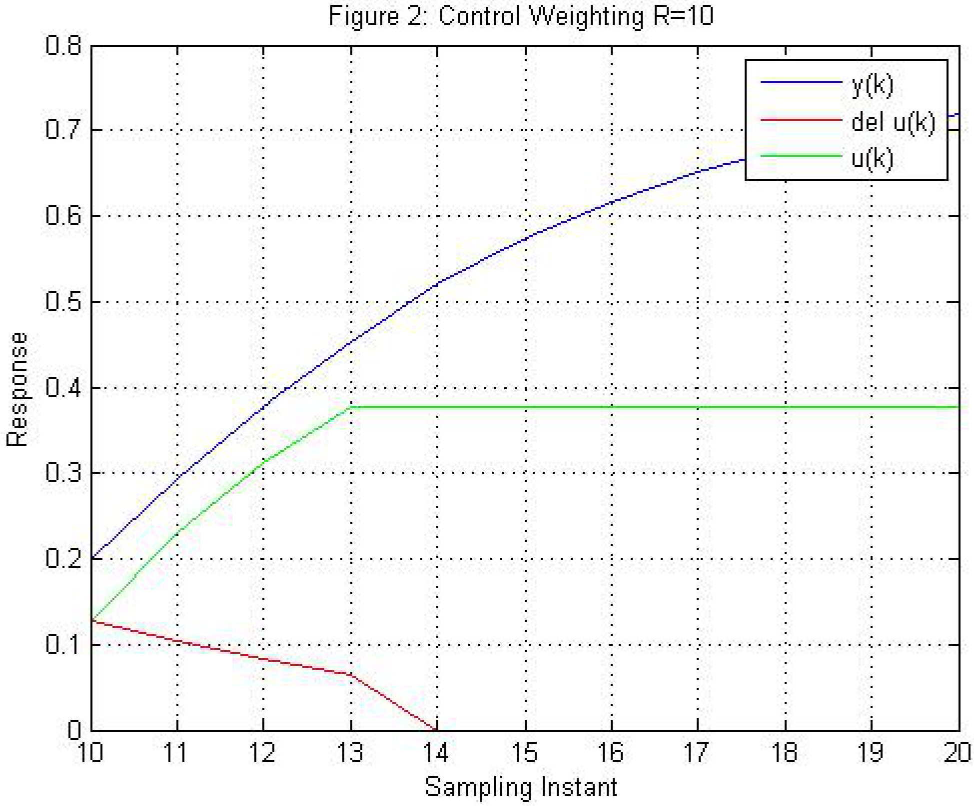
- For case 1) where $R = [0]$, the optimal input sequence is

$$\boldsymbol{\Delta U} = (\Phi^T\Phi)^{-1}(\Phi^T R_s - \Phi^T F\boldsymbol{x}(k_i)) = \begin{bmatrix} 7.2 \\ -6.4 \\ 0 \\ 0 \end{bmatrix}$$




- For case 2) where $R = 10 \cdot I$, we obtain

$$\Delta \boldsymbol{U} = (\Phi^T \Phi + 10 \cdot I)^{-1}(\Phi^T R_s - \Phi^T F \boldsymbol{x}(k_i)) = \begin{bmatrix} 0.1269 \\ 0.1034 \\ 0.0829 \\ 0.065 \end{bmatrix}$$




## Receding Horizon Control

- Although the optimal parameter vector $\boldsymbol{\Delta U}$ contains the control values $\triangle u(k_i)$, $\triangle u(k_i+1)$,$\triangle u(k_i+2)$,$\ldots$ $\triangle u(k_i+N_C-1)$ , we only implement the first sample of the sequence.
- Consider again the first-order system with state-space description

$$\boldsymbol{x}_m(k+1) = 0.8\boldsymbol{x}_m(k) + 0.1u(k)$$

- Let $r_w = 0$; initial conditions are $\boldsymbol{x}(10) = \begin{bmatrix} 0.1 & 0.2 \end{bmatrix}^t$ and $u(9) = 0$.
- As found previously, at sample time $k_i = 10$, the optimal control was found to be $\triangle u(10) = 7.2$.
- Assuming $\triangle u(9) = 0$, then we have the control signal to the plant is $u(10) = u(9) + \triangle u(10) = 7.2$.
- Compute the next plant state variable:

$$\boldsymbol{x}_m(11) = 0.8\boldsymbol{x}_m(10) + 0.1u(10) = 0.88$$

- At $k_i = 11$, new plant information is

$$\triangle \boldsymbol{x}_m(11) = 0.88 - 0.2 = 0.68$$
$$y(11) = 0.88$$

which gives

$$\boldsymbol{x}(11) = \begin{bmatrix} 0.68 \\ 0.88 \end{bmatrix}$$

- We can then calculate:

$$\triangle \boldsymbol{U} = (\Phi^T \Phi)^{-1}(\Phi^T R_s - \Phi^T F \boldsymbol{x}(11)) = \begin{bmatrix} -4.24 \\ -0.96 \\ 0 \\ 0 \end{bmatrix}$$

and thus the optimal control input $u(11) = u(10) + \triangle u(11) = 2.96$.

- Propagate the state again,

$$\boldsymbol{x}_m(12) = 0.8\boldsymbol{x}_m(11) + 0.1u(11) = 1.0$$

- At $k_i = 12$, the new plant information is

$$\triangle \boldsymbol{x}_m(12) = 1.0 - 0.88 = 0.12$$
$$y(12) = 1.0$$

which gives

$$\boldsymbol{x}(12) = \begin{bmatrix} 0.12 \\ 1.0 \end{bmatrix}$$

and

$$\triangle \boldsymbol{U} = (\Phi^T \Phi)^{-1}(\Phi^T R_s - \Phi^T F \boldsymbol{x}(12)) = \begin{bmatrix} -0.96 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Continuing,

$$\begin{aligned} \boldsymbol{x}_m(13) &= a\boldsymbol{x}_m(12) + bu(12) = 1.0 \\ y(13) &= 1.0 \end{aligned}$$

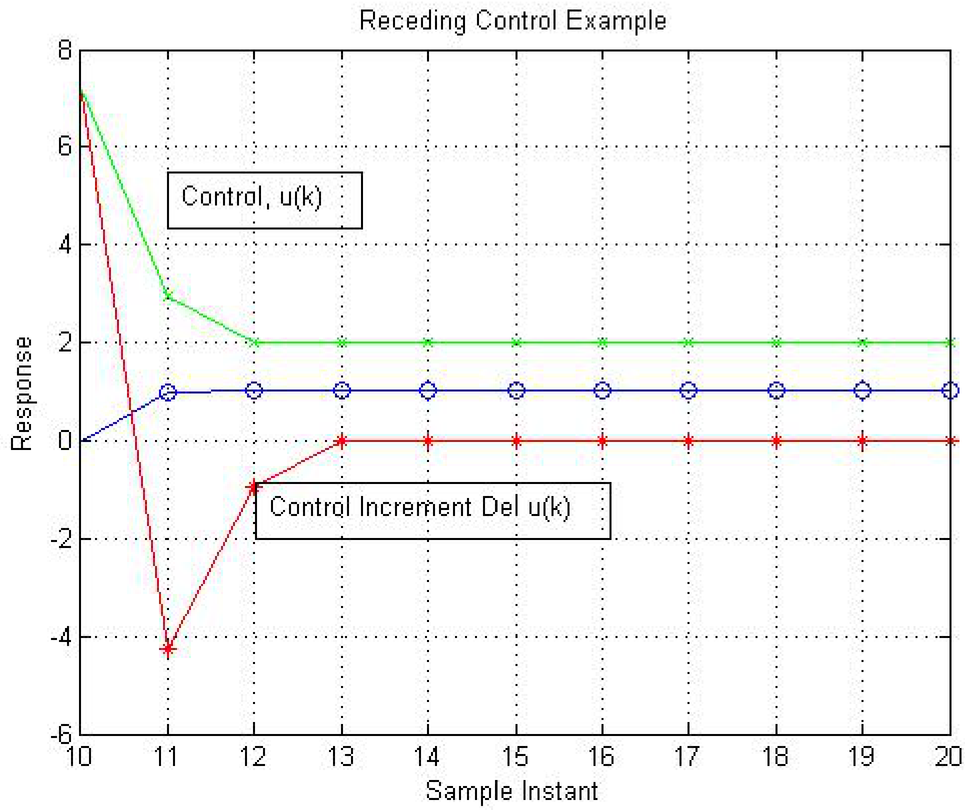
$$\begin{aligned} \triangle \boldsymbol{x}_m(13) &= 1.0 - 1.0 = 0 \\ y(13) &= 1.0 \end{aligned}$$

$$\triangle \boldsymbol{U} = (\Phi^T \Phi)^{-1}(\Phi^T R_s - \Phi^T F \boldsymbol{x}(13)) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Stacking the the computed values of $\triangle \boldsymbol{U}$ for time instants $k_i = \begin{bmatrix} 10 & 11 & 12 & 13 \end{bmatrix}$ column-wise we obtain

$$\triangle \boldsymbol{U}_{ALL} = \begin{bmatrix} 7.2 & -4.24 & -0.96 & 0 \\ -6.4 & -0.96 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where the optimal control input sequence is given by the first row. The optimal response follows:




## *Closed Loop Control System*

- We've shown that at a given time $k_i$ , the optimal control input vector $\triangle \boldsymbol{U}$ is solved from

$$\triangle \boldsymbol{U} = (\Phi^T \Phi + \bar{R})^{-1}(\Phi^T R_s - \Phi^T F \boldsymbol{x}(k_i))$$

  - Here, the first term, $(\Phi^T \Phi + \bar{R})^{-1}\Phi^T R_s$ corresponds to the set-point change
  - The second term, $(\Phi^T \Phi + \bar{R})^{-1}\Phi^T F$ corresponds to state feedback control (within the context of predictive control)

- We take just the first element of $\triangle \boldsymbol{U}$ at time $k_i$ as the incremental control,

$$\begin{aligned}\triangle \boldsymbol{U}(k_i) &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}(\Phi^T \Phi + \bar{R})^{-1}(\Phi^T \bar{R}_s r(k_i) - \Phi^T F \boldsymbol{x}(k_i)) \\ &= K_y r(k_i) - K_{mpc}\boldsymbol{x}(k_i)\end{aligned}$$

where $K_y$ is the first element of $(\Phi^T\Phi + \bar{R})^{-1}\Phi^T R_s$ and $K_{mpc}$ is the first row of $(\Phi^T\Phi + \bar{R})^{-1}\Phi^T F$ .

- These relationships depict linear time-invariant *state feedback control*.
- The closed-loop system is found by using the augmented system equation to write

$$\begin{aligned} \boldsymbol{x}(k+1) &= A\boldsymbol{x}(k) - BK_{mpc}\boldsymbol{x}(k) + BK_y r(k) \\ &= (A - BK_{mpc})\boldsymbol{x}(k) + BK_y r(k) \end{aligned}$$

  – Closed-loop eigenvalues are then the roots of the closed-loop characteristic equation:

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

## Predictive Control with Unmeasured States

### *State Estimation*

- In the design of model predictive controllers, we initally assume that the information $\boldsymbol{x}(k_i)$ is available at time $k_i$, which assumes all state variables are measurable.
- In practice this is not the case and we must instead estimate state values from the available process measurements.
  – The general construction we devise for this task is termed an *observer*.
- Assume a plant state-space model,

$$\boldsymbol{x}_m(k+1) = A_m\boldsymbol{x}_m(k) + B_m u(k)$$

- We can then calculate an estimate of the state variable, $\hat{\boldsymbol{x}}_m(k),\ k = 1, 2, \ldots$, with an initial condition $\hat{\boldsymbol{x}}_m(0)$ and input signal $u(k)$ as

$$\hat{\boldsymbol{x}}_m(k+1) = A_m\hat{\boldsymbol{x}}_m(k) + B_m u(k)$$

- What's the problem with this estimate model?
  - open loop prediction
  - works only if initial condition is accurate
  - requires stable plant - convergence only if eigenvalues of $A$ inside unit circle
  - can't regulate convergence rate
- *Better approach* $\rightarrow$ Use feedback that relies on an error measure to improve the estimate:

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

where $K_{ob}$ is an observer gain matrix.

- First term propagates the original model
- Second term is a correction
  - Correction based on the error between the measured output and the predicted value based on the state estimate.
- How do we choose the observer gain, $K_{ob}$?
  - Define the error state to be $\widetilde{\boldsymbol{x}}_m(k+1) = \boldsymbol{x}_m(k) - \hat{\boldsymbol{x}}_m(k)$; then

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

  - With an initial error state $x_m(0)$, we have

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

- Unlike the open-loop estimator case, now we can exploit the degrees of freedom in the observer gain matrix to affect the estimator convergence rate.
- One way to do this is to place the closed-loop poles of the error system matrix at desired locations in the complex plane.

## Example

- The linearized equation of motion of a simple pendulum is:

$$\frac{d^2\theta}{dt^2} + \omega^2\theta = u$$

- Design an observer that reconstructs the pendulum angle, $\theta$, from measurements of its angular velocity, ${}^{d\theta}/_{dt}$. Assume $\omega = 2$ rad/sec and sampling interval $\triangle t = 0.1$ sec.
- Select states, $x_1(t) = \theta$ and $x_2(t) = \dot{\theta}$; the continuous time state-space model is,

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

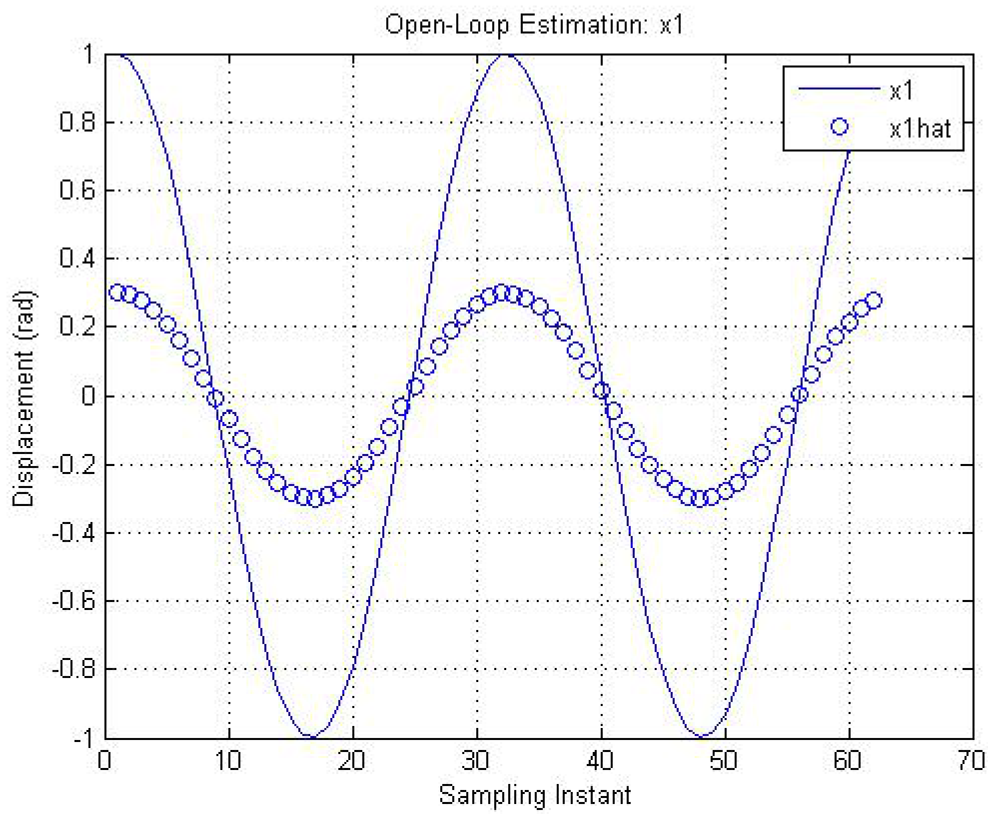
$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.$$

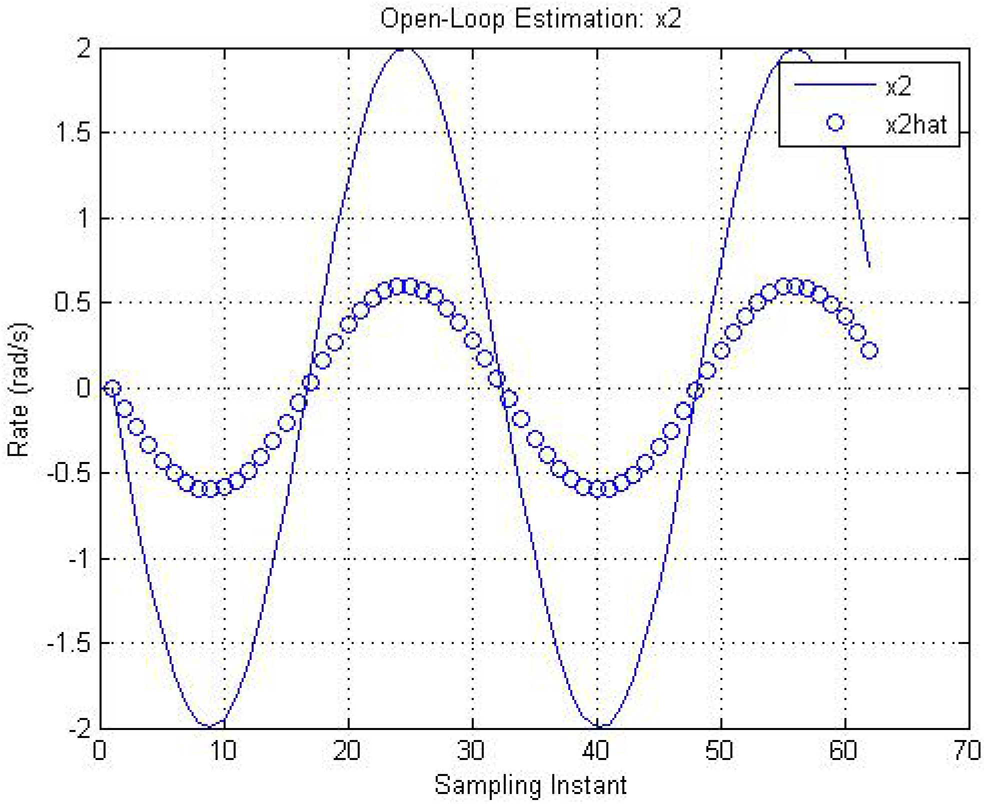
- The discrete-time model is found from the Matlab function [sysd] = c2d(sysc, .1):

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9801 & 0.0993 \\ -.3973 & 0.9801 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.0050 \\ 0.0993 \end{bmatrix} u(k)$$

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

- What if we just use the pendulum model alone for (open-loop) prediction?
  - Assume $u(k) = 0$, and the initial conditions, $\theta(0) = x_1(0) = 1$ and $\dot{\theta}(0) = x_2(0) = 0$.
  - Now let's assume we take a guess at the initial conditions of the estimator as $\hat{x}_1(0) = 0.3$ and $\hat{x}_2(0) = 0$.
  - Results are as follows:






– It's easy to see that the open-loop estimation was unable to recover from an error in the initial conditions.

- Now let us design and implement an observer to predict the angle of the pendulum.

  – Assume the observer gain is given by $K_{ob} = \begin{bmatrix} \kappa_1 & \kappa_2 \end{bmatrix}^t$.

  – The close-loop characteristic polynomial of the observer is

$$\det\left(\lambda I - [A_m - K_{ob}C_m]\right) = (\lambda - 0.9801)(\lambda + \kappa_2 - .9801) - .3973 \times (\kappa_1 - .0993)$$

  – Let's set the observer poles at $0.1$ and $0.2$ for good convergence speed.

  – Equating the above expression to the desired closed cloop characteristic equation for the observer, namely
$(\lambda - 0.9801)(\lambda + \kappa_2 - .9801) - .3973 \times (\kappa_1 - .0993) = (\lambda - 0.1)(\lambda - 0.2)$

we can solve for the unknown observer gains

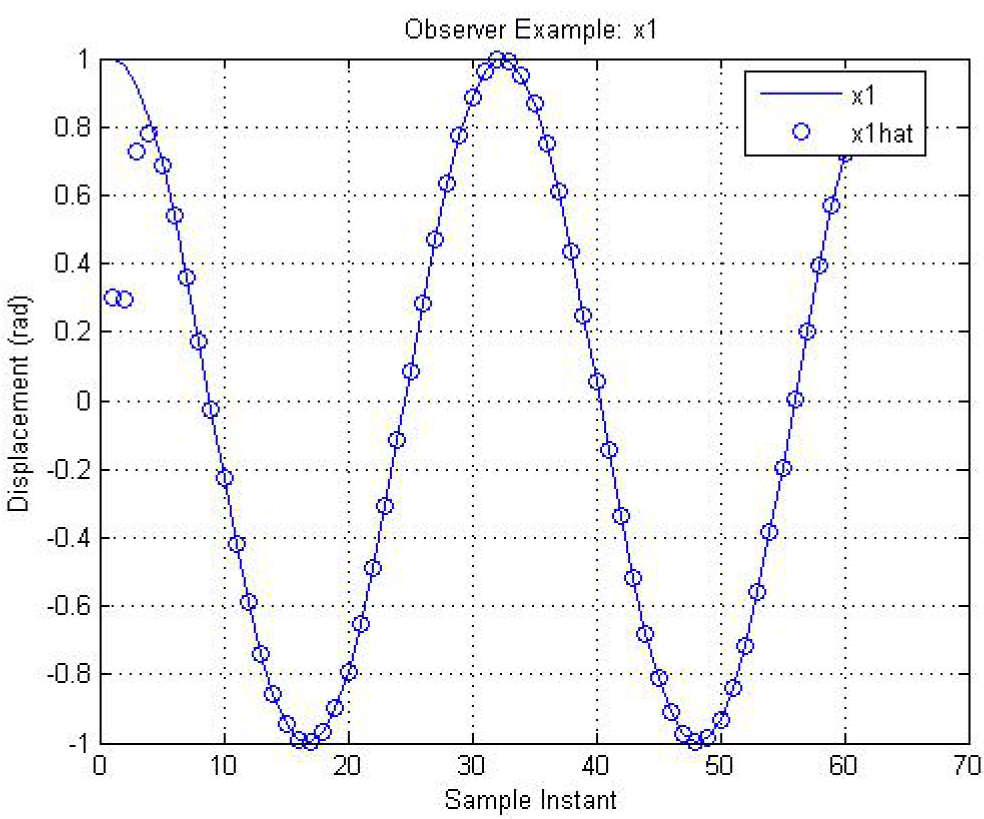
$$\begin{bmatrix} 0 & 1 \\ -.3973 & -.9801 \end{bmatrix} \begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix} = \begin{bmatrix} 1.6601 \\ -0.9800 \end{bmatrix}$$

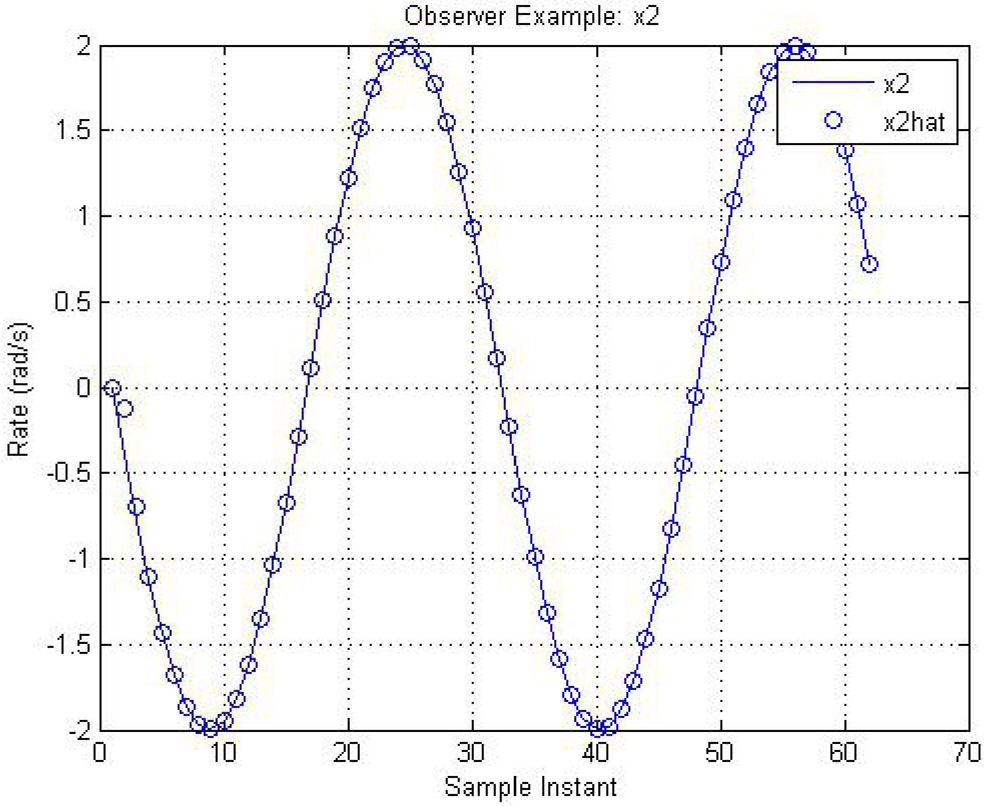
$$\begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix} = \begin{bmatrix} -1.6284 \\ 1.6601 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ -.3973 & -.9801 \end{bmatrix} \begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix} = \begin{bmatrix} 1.6601 \\ -0.9800 \end{bmatrix}$$

$$\begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix} = \begin{bmatrix} -1.6284 \\ 1.6601 \end{bmatrix}$$

– Implementing the estimator with the computed observer gives the results depicted below.






– Here we see the observer quickly converges to the true value in approximately three time steps.

## Predictive Control Using State Estimates

- In cases where the state variable $x(k_i)$ at time $k_i$ is not measurable, we use an observer in the implementation of predictive control.

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

- Note that the state matrices, $\left( A,\ B,\ C \right)$ represent the augmented model used for predictive controller design.

- The corresponding modified cost function $J$ can then be written

$$J = (R_s - F\hat{x}(k_i))^T(\bar{R}_s r(k_i) - F\hat{x}(k_i)) - 2\triangle U^T\Phi^T(R_s - F\hat{x}(k_i)) + \triangle U^T(\Phi^T\Phi -$$

– The optimal solution is thus:

$$\triangle U^* = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - F\hat{\boldsymbol{x}}(k_i))$$

- Application of receding horizon control leads to the optimal solution at time $k_i$:

$$\triangle u(k_i) = K_y r(k_i) - K_{mpc}\hat{\boldsymbol{x}}(k_i)$$

  – This is standard state feedback control with estimate $\hat{\boldsymbol{x}}(k_i)$.

- Let's now examine the close-loop properties of this system.
- First, for the closed-loop state equations,

$$\begin{aligned} \boldsymbol{x}(k+1) &= A\boldsymbol{x}(k) + B\triangle u(k) \\ &= A\boldsymbol{x}(k) + BK_y r(k) - BK_{mpc}\hat{\boldsymbol{x}}(k) \end{aligned}$$

- The closed-loop observer error equation is:

$$\tilde{\boldsymbol{x}}(k+1) = (A - K_{ob}C)\tilde{\boldsymbol{x}}(k)$$

  where

$$\tilde{\boldsymbol{x}}(k) = \boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k)$$

- Substituting for $\hat{\boldsymbol{x}}(k)$ we can write:

$$\boldsymbol{x}(k+1) = (A - BK_{mpc})\boldsymbol{x}(k) - BK_{mpc}\tilde{\boldsymbol{x}}(k) + BK_y r(k)$$

- Combining the error and system state equations,

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

  – Since the system matrix is lower block diagonal, the eigenvalues are given by the eigenvalues of the diagonal blocks, $A - K_{ob}C$ and $A - BK_{mpc}$ respectively (separation principle).

## Example

The augmented model for a double integrator plant is given by

$$\begin{aligned} \boldsymbol{x}(k+1) &= A\boldsymbol{x}(k) + B\triangle u(k) \\ y(k) &= C\boldsymbol{x}(k) \end{aligned}$$

where

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

- Design a state estimate predictive control system and simulate the closed-loop response for a set-point change.
- Design specifications:

$$\begin{aligned} N_c &= 5 \\ N_p &= 30 \\ r_w &= 10 \end{aligned}$$

- Place the observer poles at: $0.01$, $0.0105$, $0.011$ to give fast dynamic response.
- The open-loop plant has three eigenvalues at one (by inspection), two from the double-integrator and one from the embedded integrator structure. Using the Matlab function `place`, we find the observer gains from:

$$\texttt{Pole} = \begin{bmatrix} \texttt{0.01} & \texttt{0.0105} & \texttt{0.011} \end{bmatrix};$$

$$\texttt{Kob} = \texttt{place}\left( \texttt{A}' \ \texttt{C}' \ \texttt{Pole} \right)';$$

which gives,

$$\texttt{Kob} = \begin{bmatrix} 1.9685 \\ 0.9688 \\ 2.9685 \end{bmatrix}$$

- Using the given performance parameters, the state feedback controller gain is computed to be,

$$\texttt{Kmpc} = \begin{bmatrix} 0.8984 & 1.3521 & 0.4039 \end{bmatrix}$$

- The resulting closed-loop eigenvalues are at $0.3172 \pm j0.4089$ and $0.3624$.

(mostly blank)

(mostly blank)

(mostly blank)