LECTURE 8

DESIGN OF DISCRETE TIME CONTROLLER — STATE SPACE APPROACH

OUTLINE

- State Feedback
  - Pole (eigenvalue) placement, when can it be done?
  - Deadbeat control
  - Other issues (input scaling, observability)
  - Comparison with output feedback
- Observer design

STATE SPACE DESIGN

- Work with State Space Equations

Continuous-Time Systems	Discrete-Time Systems
$\dot{\mathbf{x}}(t) = \mathbf{A}\cdot\mathbf{x}(t) + \mathbf{B}\cdot\mathbf{u}(t)$	$\mathbf{x}(k+1) = \mathbf{A}\cdot\mathbf{x}(k) + \mathbf{B}\cdot\mathbf{u}(k)$
$\mathbf{y}(t) = \mathbf{C}\cdot\mathbf{x}(t) + \mathbf{D}\cdot\mathbf{u}(t)$	$\mathbf{y}(k) = \mathbf{C}\cdot\mathbf{x}(k) + \mathbf{D}\cdot\mathbf{u}(k)$

$$x: \quad n\times 1$$

$$u: \quad r\times 1$$

$$y: \quad m\times 1$$

- Less “graphical”, more computational (eigenvalue, norm, cost function)
- Mostly result in full-state feedback algorithms

STATE FEEDBACK

- Assumes that all the state variables are available (from direct measurement, estimation, observation, etc.) .

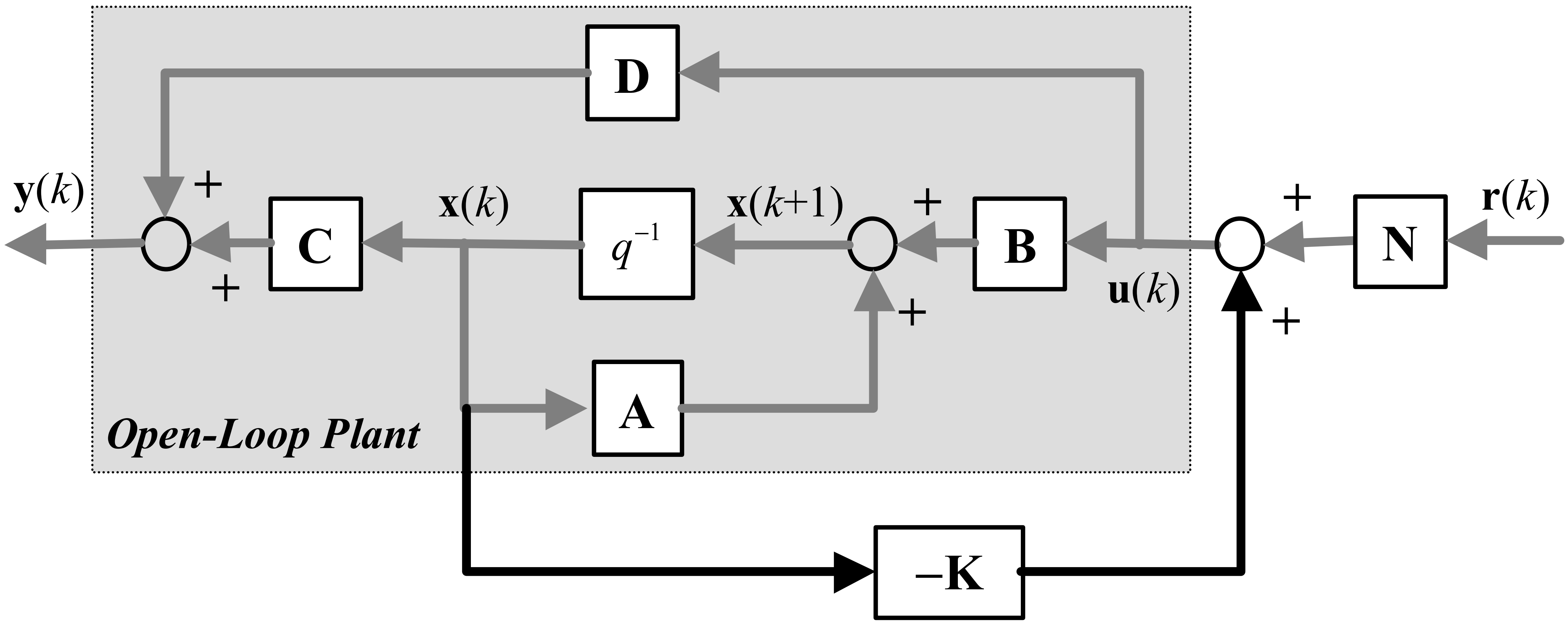
<u>Continuous-Time Systems</u> <u>Discrete-Time Systems</u>

$$\mathbf{u}(t) = -\underbrace{\mathbf{K}}_{\textit{state feedback gain}} \cdot \mathbf{x}(t) + \mathbf{N} \cdot \mathbf{r}(t) \qquad \mathbf{u}(k) = -\mathbf{K} \cdot \mathbf{x}(k) + \underbrace{\mathbf{N}}_{\textit{feedforward gain}} \cdot \mathbf{r}(k)$$

$$\dot{\mathbf{x}}(t) = \left(\mathbf{A} - \mathbf{BK}\right) \cdot \mathbf{x}(t) + \mathbf{BN} \cdot r(t) \qquad \mathbf{x}(k+1) = \left(\mathbf{A} - \mathbf{BK}\right) \cdot \mathbf{x}(k) + \mathbf{BN} \cdot r(k)$$

$$\mathbf{y}(t) = \left(\mathbf{C} - \mathbf{DK}\right) \cdot \mathbf{x}(t) + \mathbf{DN} \cdot r(t) \qquad \mathbf{y}(k) = \left(\mathbf{C} - \mathbf{DK}\right) \cdot \mathbf{x}(k) + \mathbf{DN} \cdot r(k)$$

STATE FEEDBACK (CONT.)



Note: the feedback law is usually written as $\boxed{\mathbf{u} = -\mathbf{K} \cdot \mathbf{x}}$ in the literature. This is based on the assumption of a "regulation" problem. In other words, the output **y** is controlled to stay close to zero. The configuration above is for a general (tracking) problem where the reference signal **r** may not be zero.

POLE PLACEMENT THROUGH STATE FEEDBACK

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK}) \cdot \mathbf{x}(t) + \mathbf{BN} \cdot r(t) \qquad \mathbf{x}(k+1) = (\mathbf{A} - \mathbf{BK}) \cdot \mathbf{x}(k) + \mathbf{BN} \cdot r(k)$$

$$\mathbf{y}(t) = (\mathbf{C} - \mathbf{DK}) \cdot \mathbf{x}(t) + \mathbf{DN} \cdot r(t) \qquad \mathbf{y}(k) = (\mathbf{C} - \mathbf{DK}) \cdot \mathbf{x}(k) + \mathbf{DN} \cdot r(k)$$

Fact: If (A,B) is in controllable canonical form, pole placement is possible $\Rightarrow$ If (A,B) controllable, can be transformed to CCF, pole placement is possible.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \cdot u(k)$$

$$\begin{aligned} u(k) &= -\begin{bmatrix} k_n & k_{n-1} & k_{n-2} & \cdots & k_1 \end{bmatrix} \cdot \mathbf{x}(k) + N \cdot r(k) \\ &= -k_n \cdot x_1(k) - k_{n-1} \cdot x_2(k) - \cdots - k_1 \cdot x_n(k) + N \cdot r(k) \end{aligned}$$

POLE PLACEMENT (CONT.)

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \cdot u(k)$$

$$\begin{aligned} u(k) &= -\begin{bmatrix} k_n & k_{n-1} & k_{n-2} & \cdots & k_1 \end{bmatrix} \cdot \mathbf{x}(k) + N \cdot r(k) \\ &= -k_n \cdot x_1(k) - k_{n-1} \cdot x_2(k) - \cdots - k_1 \cdot x_n(k) + N \cdot r(k) \end{aligned}$$

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -(k_n + a_n) & -(k_{n-1} + a_{n-1}) & -(k_{n-2} + a_{n-2}) & \cdots & -(k_1 + a_1) \end{bmatrix} \mathbf{x}(k) + BN \cdot r(k)$$

Closed-loop char. eq.

$$z^n + (k_1 + a_1) z^{n-1} + (k_2 + a_2) z^{n-2} + \cdots + (k_n + a_n) = 0$$

EX8.1 STATE FEEDBACK POLE PLACEMENT

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-k}{m} & \frac{-b}{m} \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f$$

State feedback:

$$\det\left[s\mathbf{I} - \mathbf{A} + \mathbf{BK}\right] = \det\left[\begin{bmatrix} s & -1 \\ \frac{k}{m} & s + \frac{b}{m} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}[k_2 \ \ k_1]\right] = \det\begin{bmatrix} s & -1 \\ \frac{k+k_2}{m} & s + \frac{b+k_1}{m} \end{bmatrix}$$

$$= s\left(s + \frac{b+k_1}{m}\right) + \frac{k+k_2}{m} = s^2 + \frac{b+k_1}{m}s + \frac{k+k_2}{m} = 0$$

If the desired locations of the closed-loop poles are $-\alpha_1$ and $-\alpha_2$

$$\frac{b+k_1}{m} = \alpha_1 + \alpha_2 \quad \Rightarrow \quad k_1 = m(\alpha_1 + \alpha_2) - b$$

$$\frac{k+k_2}{m} = \alpha_1\alpha_2 \quad \Rightarrow \quad k_2 = m\alpha_1\alpha_2 - k$$

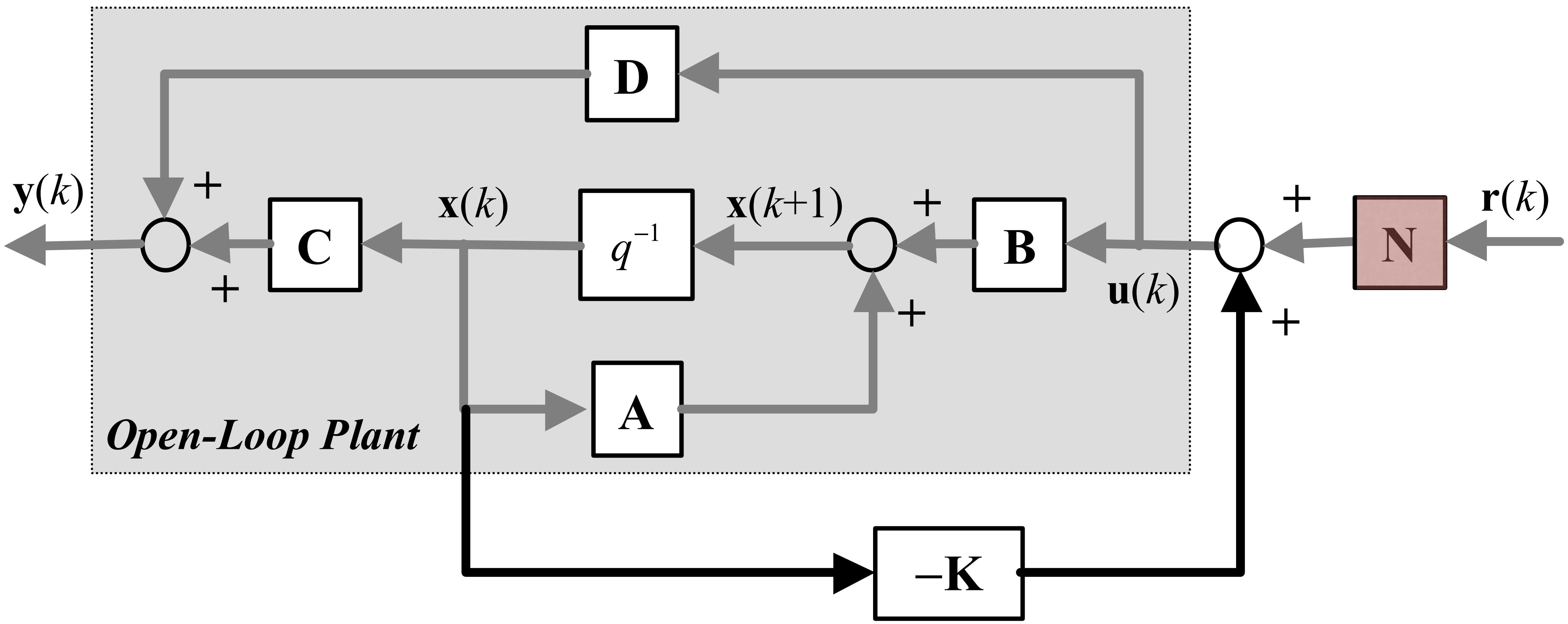
MATLAB COMMANDS

```
» help place

 PLACE  Pole placement technique

    K = PLACE(A,B,P)  computes a state-feedback matrix K such that
    the eigenvalues of  A-B*K  are those specified in vector P.
    No eigenvalue should have a multiplicity greater than the
    number of inputs.

    [K,PREC,MESSAGE] = PLACE(A,B,P)  returns PREC, an estimate
of how
    closely the eigenvalues of A-B*K match the specified locations P
    (PREC measures the number of accurate decimal digits in the
actual
    closed-loop poles).  If some nonzero closed-loop pole is more than
    10% off from the desired location, MESSAGE contains a warning
    message.
```

REFERENCE INPUT SCALING

State feedback can change the characteristic polynomial of the system, therefore, the steady state gain of the closed-loop system is changed. It is necessary to adjust feedforward gain **N**.

Usually the feedforward gain **N** is selected to make the overall closed-loop system steady-state gain equals to one.



REFERENCE INPUT SCALING

For a strictly causal, SISO system

Continuous-Time Systems

$$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{b} \cdot u(t)$$

$$y(t) = \mathbf{C} \cdot \mathbf{x}(t)$$

$$\mathbf{u}(t) = -\mathbf{K} \cdot \mathbf{x}(t) + \mathbf{N} \cdot \mathbf{r}(t)$$

Discrete-Time Systems

$$\mathbf{x}(k+1) = \mathbf{A} \cdot \mathbf{x}(k) + \mathbf{b} \cdot u(k)$$

$$y(k) = \mathbf{C} \cdot \mathbf{x}(k)$$

$$\mathbf{u}(k) = -\mathbf{K} \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{r}(k)$$

Closed-loop (from *r* to *y*)

$$\boxed{\mathbf{G}_{CL}(s) = \mathbf{C}\left[s\mathbf{I} - \mathbf{A} + \mathbf{BK}\right]^{-1}\mathbf{B} \cdot N}$$

$$N = \left(\mathbf{C}\left[-\mathbf{A} + B\mathbf{K}\right]^{-1} B\right)^{-1}$$

$$= \frac{1}{\mathbf{C}\left[-\mathbf{A} + B\mathbf{K}\right]^{-1} B}$$

$$\boxed{\mathbf{G}_{CL}(z) = \mathbf{C}\left[z\mathbf{I} - \mathbf{A} + \mathbf{BK}\right]^{-1}\mathbf{B} \cdot N}$$

$$N = \left(\mathbf{C}\left[\mathbf{I} - \mathbf{A} + B\mathbf{K}\right]^{-1} B\right)^{-1}$$

$$= \frac{1}{\mathbf{C}\left[\mathbf{I} - \mathbf{A} + B\mathbf{K}\right]^{-1} B}$$

EXAMPLE FB/FF SIMULATIONS

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} u(k)$$

$$u(k) = -\begin{bmatrix} k_2 & k_1 \end{bmatrix} \cdot \mathbf{x}(k)$$

Desired char. eq.

$$z^2 + p_1 z + p_2 = 0 \quad \Rightarrow \quad \boxed{\begin{aligned} k_1 &= \frac{1}{2T}(3 + p_1 - p_2) \\ k_2 &= \frac{1}{T^2}(1 + p_1 + p_2) \end{aligned}}$$

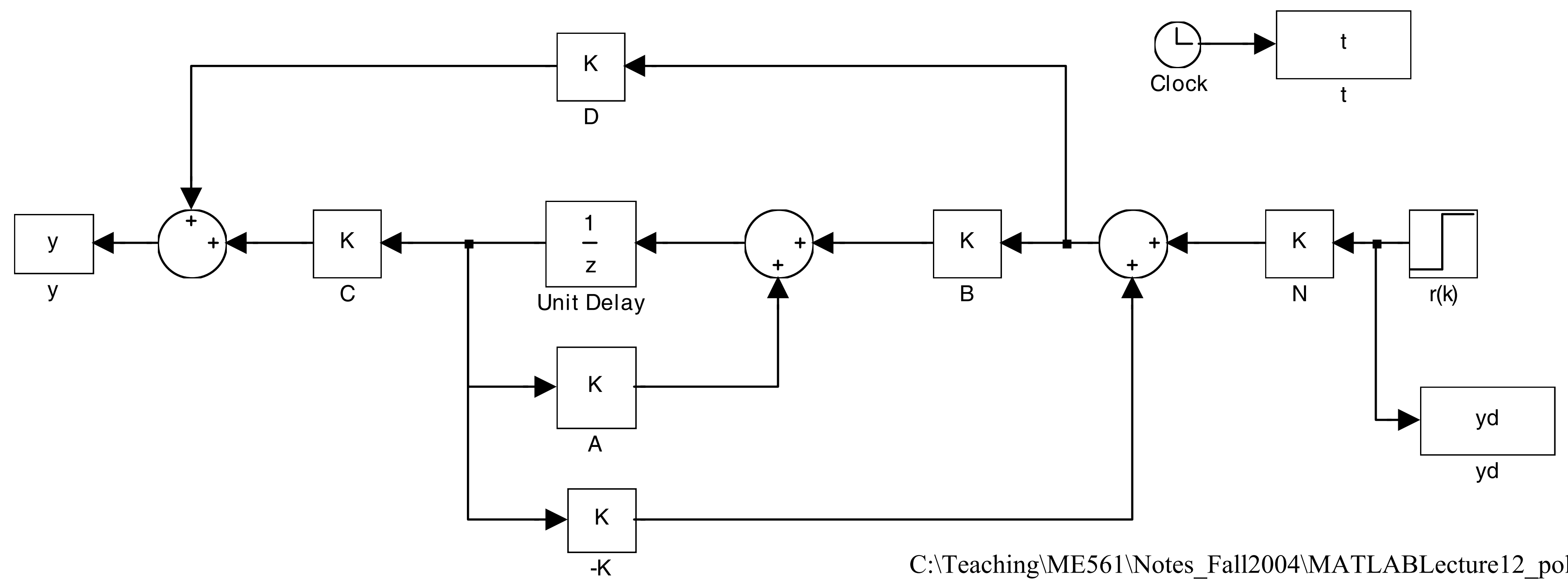
Pole selection

$$p_1 = -2\cos\left(\omega_n{}^* T\sqrt{1-\zeta^{*2}}\right) \cdot e^{-\zeta^* \omega_n^* T}$$

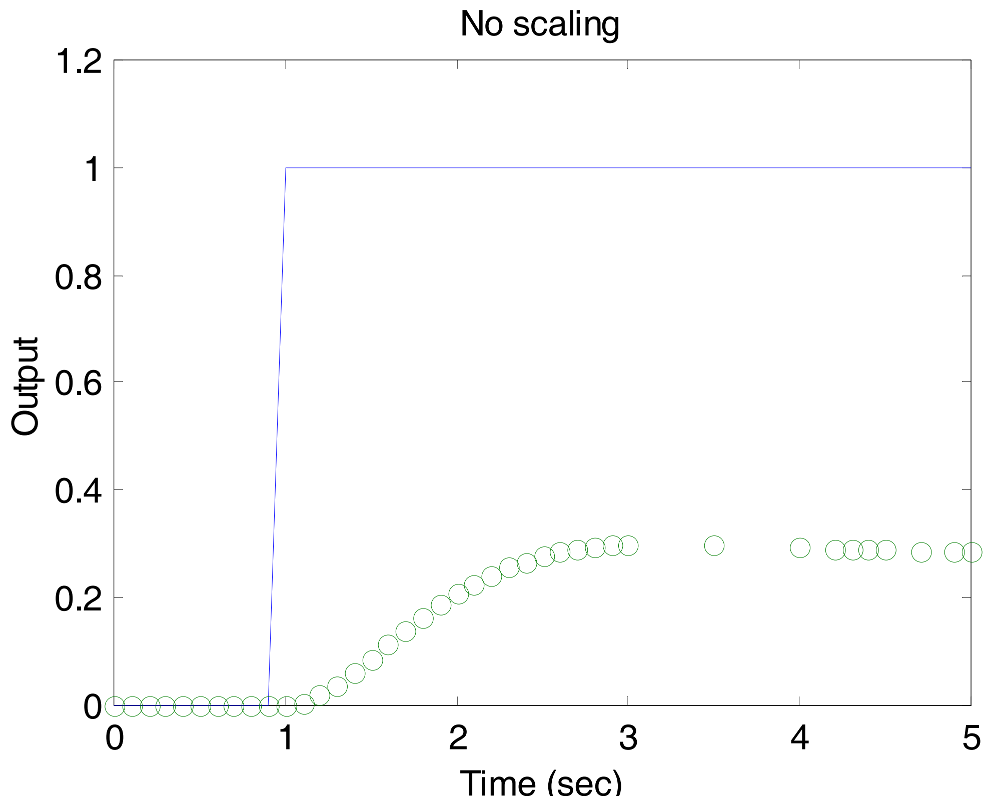
$$p_2 = e^{-2\zeta^* \omega_n^* T}$$

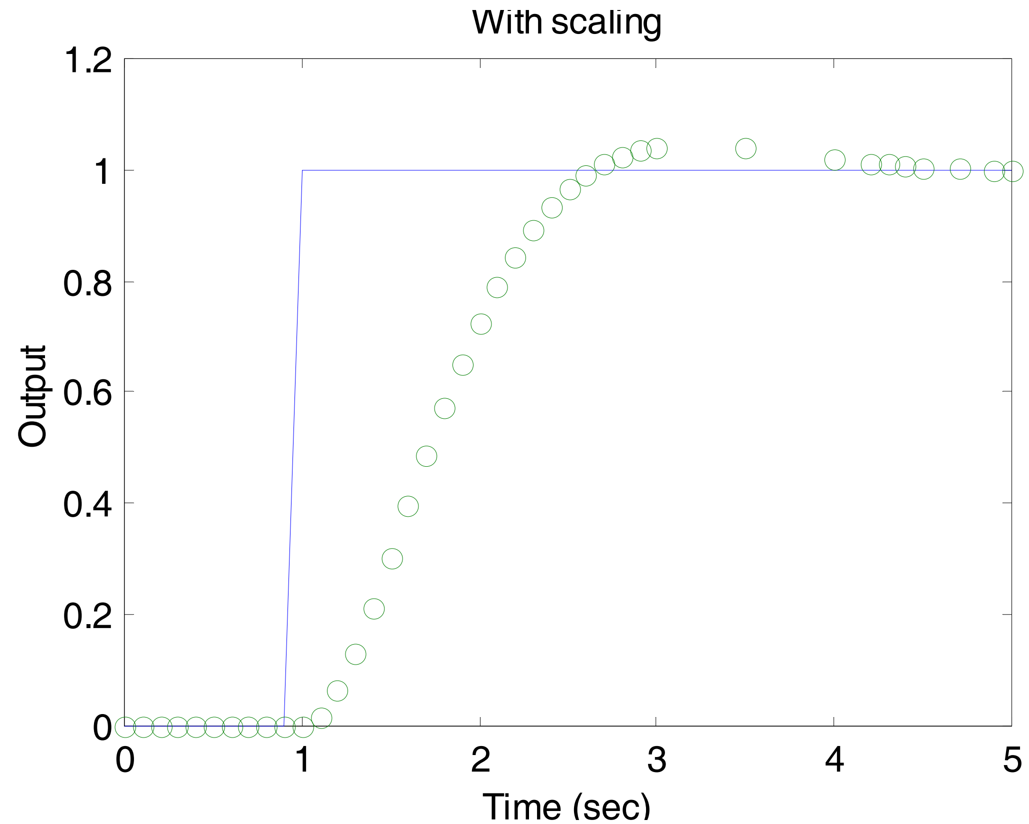
$$\begin{aligned} \zeta^* &= 0.7 \\ \omega_n^* &= 2.0 \\ T &= 0.1 \end{aligned} \quad \Rightarrow \quad \begin{aligned} p_1 &= -1.721 \\ p_2 &= 0.7558 \end{aligned} \quad \Rightarrow \quad \begin{aligned} k_1 &= 2.616 \\ k_2 &= 3.4772 \end{aligned}$$

EXAMPLE (CONT.)



C:\Teaching\ME561\Notes_Fall2004\MATLABLecture12_pole_placement





DEADBEAT (FINITE SETTLING TIME) CONTROL

All the closed-loop poles are placed at the origin, i.e., closed-loop char. eq.

$$z^n = 0$$

No matter how large the initial condition **x**(0) may be, the system will be brought to the origin in *n* steps.

Proof: $\mathbf{x}(k+1) = (\mathbf{A} - \mathbf{BK}) \cdot \mathbf{x}(k)$

From the Cayley-Hamilton theorem, we have $(\mathbf{A} - \mathbf{BK})^n = \mathbf{0}$

$\Rightarrow$ $\mathbf{x}(n) = (\mathbf{A} - \mathbf{BK})^n \cdot \mathbf{x}(0) = 0$

EX8.4 DEADBEAT CONTROL

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} u(k)$$

$$u(k) = -\begin{bmatrix} k_2 & k_1 \end{bmatrix} \cdot \mathbf{x}(k)$$

Desired char. eq.

$$z^2 + p_1 z + p_2 = 0 \quad \Rightarrow \quad \boxed{\begin{aligned} k_1 &= \frac{1}{2T}(3 + p_1 - p_2) \\ k_2 &= \frac{1}{T^2}(1 + p_1 + p_2) \end{aligned}}$$

Deadbeat:

$$k_1 = \frac{3}{2T} \quad \text{and} \quad k_2 = \frac{1}{T^2}$$

$$\mathbf{x}(k+1) = \begin{bmatrix} \frac{1}{2} & \frac{T}{4} \\ -\frac{1}{T} & \frac{-1}{2} \end{bmatrix} \mathbf{x}(k) = \mathbf{A}_{CL} \cdot \mathbf{x}(k) \qquad \mathbf{A}_{CL}^2 = \begin{bmatrix} \frac{1}{2} & \frac{T}{4} \\ -\frac{1}{T} & \frac{-1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{T}{4} \\ -\frac{1}{T} & \frac{-1}{2} \end{bmatrix} = \mathbf{0}$$

OTHER ISSUES

- Controllability is not influenced by state feedback

  (assumption: N is non-singular)

$$
\begin{aligned}
Rank(\mathbf{W_C}) &= Rank\left(\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B}\end{bmatrix}\right) \\
&= Rank\left(\begin{bmatrix}\mathbf{BN} & (\mathbf{A}-\mathbf{BK})\mathbf{BN} & (\mathbf{A}-\mathbf{BK})^2\mathbf{BN} & \cdots & (\mathbf{A}-\mathbf{BK})^{n-1}\mathbf{BN}\end{bmatrix}\right)
\end{aligned}
$$

- Observability may be lost,
  - Why?
  - Why do we care?

OUTPUT FEEDBACK

$$\mathbf{x}(k+1) = \mathbf{A} \cdot \mathbf{x}(k) + \mathbf{b} \cdot u(k)$$

$$y(k) = \mathbf{C} \cdot \mathbf{x}(k)$$

$$\mathbf{u} = -\mathbf{K} \cdot \mathbf{y} + \mathbf{N} \cdot \mathbf{r}$$

Each “K” can be thought as resulting in one DOF on the root locus.

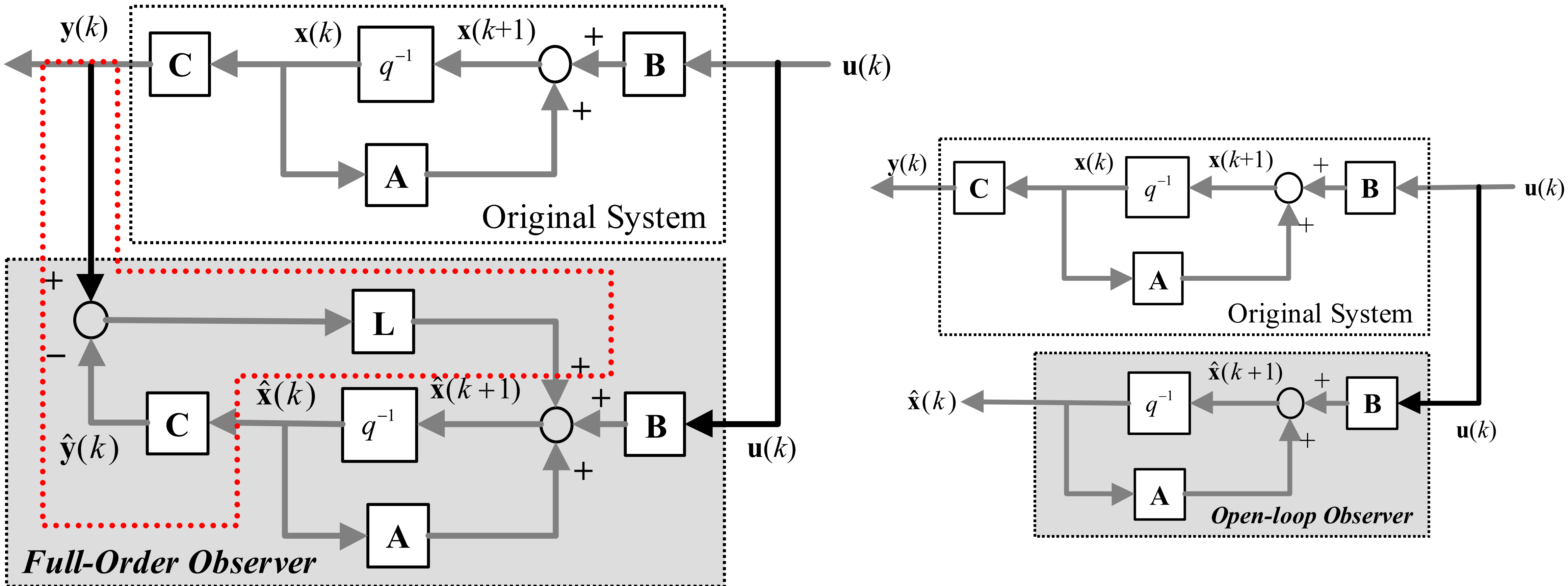
$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} \cdot u(t)$$

$$u^*(t) = -k_1 \cdot x_1(t) - k_2 \cdot x_2(t)$$

CLOSED-LOOP (LUENBERGER) OBSERVER

Luenberger Observer:

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\cdot\hat{\mathbf{x}}(k) + \mathbf{B}\cdot\mathbf{u}(k) + \mathbf{L}\cdot\left[\mathbf{y}(k) - \mathbf{C}\cdot\hat{\mathbf{x}}(k)\right]$$



POLE PLACEMENT FOR LUENBERGER OBSERVERS

- If the linear time-invariant system (A, C) is observable, eigenvalues of $\mathbf{A}_{FO}$ =A-LC can be arbitrarily assigned—dual property of full state feedback pole placement problem

$$\underbrace{\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix}}_{\mathbf{x}_O(k+1)} = \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & -a_n \\ 1 & 0 & \cdots & 0 & 0 & -a_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & -a_2 \\ 0 & 0 & \cdots & 0 & 1 & -a_1 \end{bmatrix}}_{\mathbf{A}_O} \underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix}}_{\mathbf{x}_O(k)} + \underbrace{\begin{bmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_2 \\ b_1 \end{bmatrix}}_{\mathbf{B}_O} \cdot u(k)$$

$$y(k) = \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}}_{\mathbf{C}_O} \cdot \mathbf{x}_O(k)$$

$$A - LC = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & -a_n \\ 1 & 0 & \cdots & 0 & 0 & -a_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & -a_2 \\ 0 & 0 & \cdots & 0 & 1 & -a_1 \end{bmatrix} - \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_{n-1} \\ L_n \end{bmatrix} \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & -a_n - L_1 \\ 1 & 0 & \cdots & 0 & 0 & -a_{n-1} - L_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & -a_2 - L_{n-1} \\ 0 & 0 & \cdots & 0 & 1 & -a_1 - L_n \end{bmatrix}$$

EX8.7 VELOCITY ESTIMATION (POLE PLACEMENT)

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \mathbf{x}(k) + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \cdot u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \mathbf{x}(k)$$

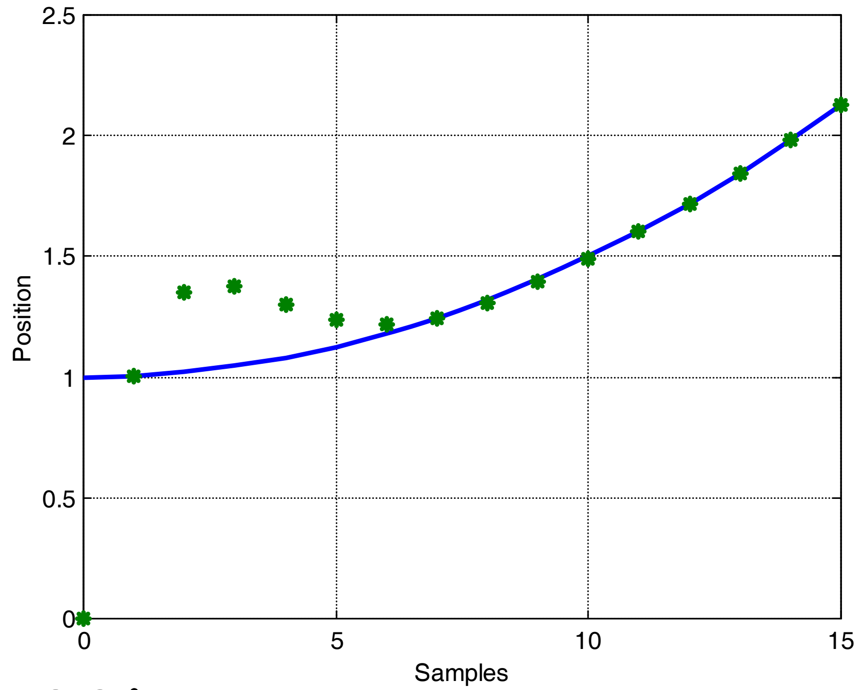
» help place

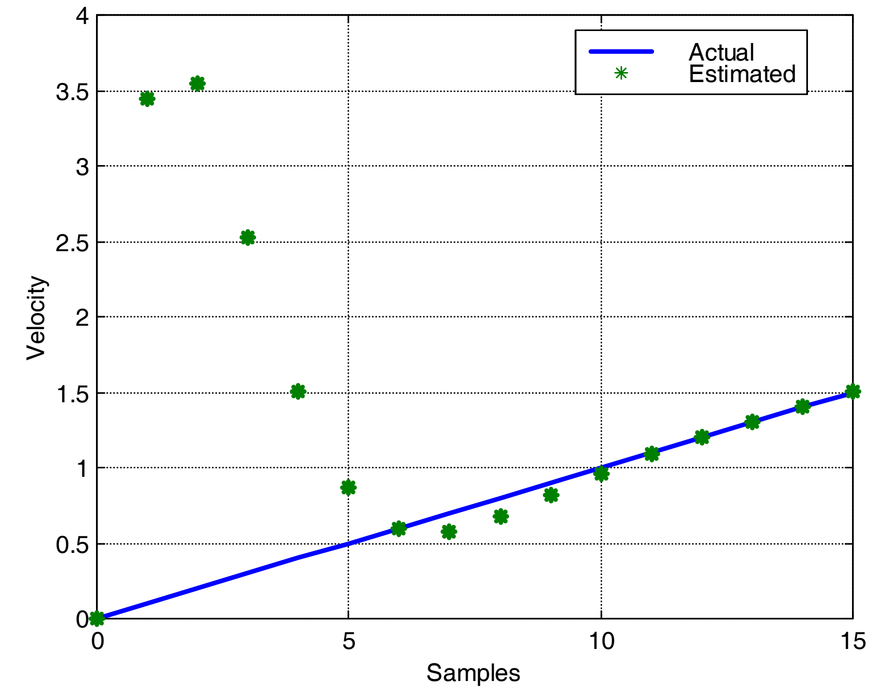
PLACE Pole placement technique

K = PLACE(A,B,P) computes a state-feedback matrix K such that the eigenvalues of A-B*K are those specified in vector P.

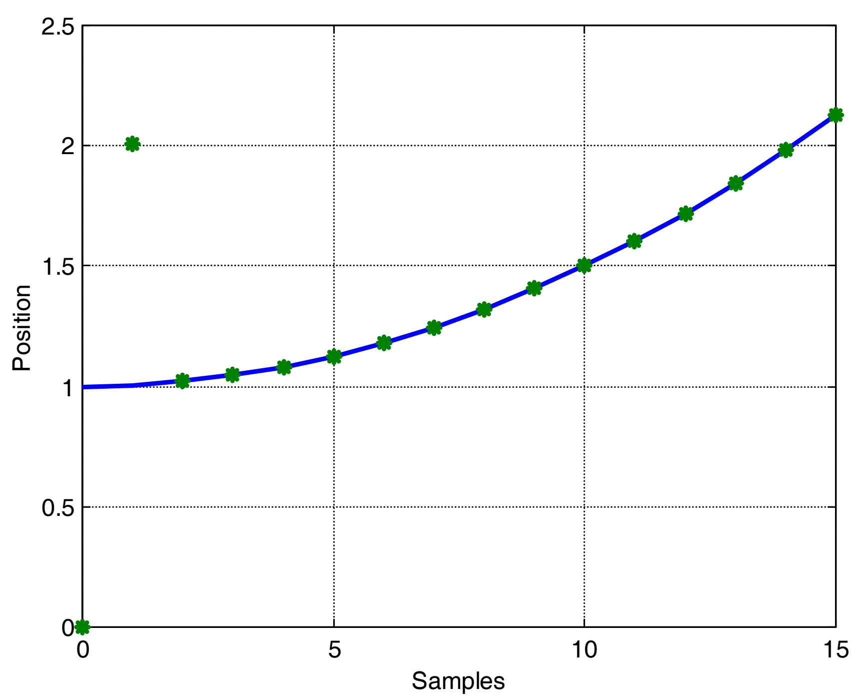
```
P=[0.5+0.29*j, 0.5-0.29*j];
T=0.1;
A=[1, T; 0, 1];  C=[1, 0];
K = place(A', C', P);
L=K';
```

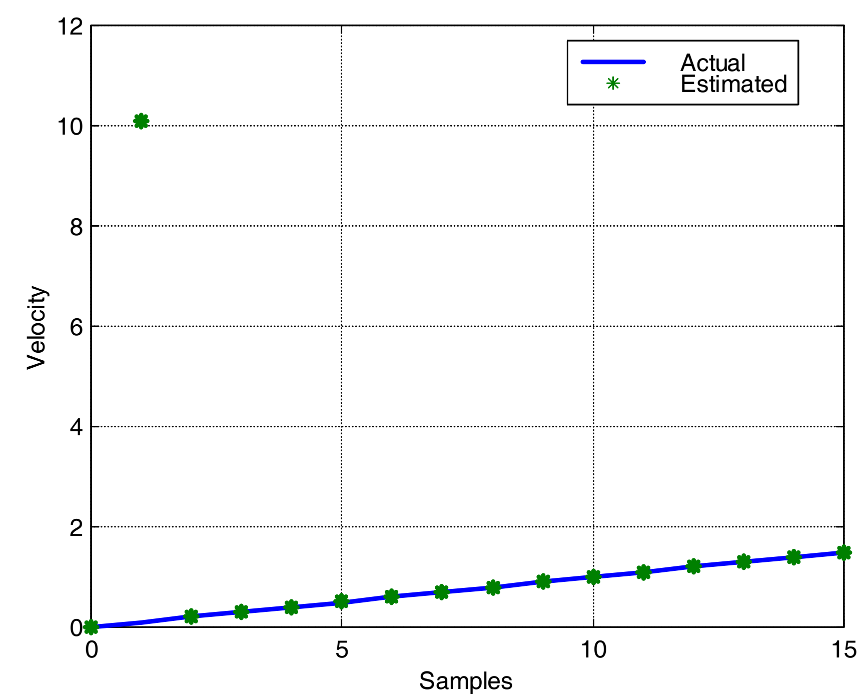
Ex8.7 (cont.)





0.5+-0.29j





0, 0

STATE PREDICTION AND CORRECTION

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\cdot\hat{\mathbf{x}}(k) + \mathbf{B}\cdot\mathbf{u}(k) + \mathbf{L}\cdot\left[\mathbf{y}(k) - \mathbf{C}\cdot\hat{\mathbf{x}}(k)\right]$$

y(k) is used to estimate (predict) x(k+1).

Can we use y(k+1) to estimate x(k+1)? Yes, add a correction step.

$\hat{\mathbf{x}}(k|k)$ Estimation of **x**(*k*) base on **y**(*l*), *l=0,…k*.

$\hat{\mathbf{x}}(k|k-1)$ Estimation of **x**(*k*) base on **y**(*l*), *l=0,…k-1*.

Predictor : $\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\cdot\hat{\mathbf{x}}(k|k) + \mathbf{B}\cdot\mathbf{u}(k)$

Corrector : $\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{L}\cdot\left[\mathbf{y}(k+1) - \mathbf{C}\cdot\hat{\mathbf{x}}(k+1|k)\right]$

$$\hat{\mathbf{x}}(k+1|k+1) = \mathbf{A}\cdot\hat{\mathbf{x}}(k|k) + \mathbf{B}\cdot\mathbf{u}(k) + \mathbf{L}\cdot\left[\mathbf{y}(k+1) - \mathbf{C}\cdot\hat{\mathbf{x}}(k+1|k)\right]$$

STATE PREDICTION AND CORRECTION (CONT.)

$$\hat{\mathbf{x}}(k+1) = \mathbf{A} \cdot \hat{\mathbf{x}}(k) + \mathbf{B} \cdot \mathbf{u}(k) + \mathbf{L} \cdot \left[\mathbf{y}(k) - \mathbf{C} \cdot \hat{\mathbf{x}}(k) \right]$$

<u>Two-step Procedure</u>

Predictor:

$$\boxed{\hat{\mathbf{x}}(k+1|k) = \mathbf{A} \cdot \hat{\mathbf{x}}(k|k) + \mathbf{B} \cdot \mathbf{u}(k)}$$

First, run openloop (based on model) to predict *x* at *k+1.*

Corrector:

$$\boxed{\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{L} \cdot \left[\mathbf{y}(k+1) - \mathbf{C} \cdot \hat{\mathbf{x}}(k+1|k) \right]}$$

Then, when y(k+1) becomes available, add correction
to obtain the updated (supposedly more accurate) *x* at *k+1.*

$$u(k) = -\mathbf{K} \cdot \hat{\mathbf{x}}(k \mid k) + Nr$$

ERROR DYNAMICS

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\cdot\hat{\mathbf{x}}(k|k) + \mathbf{B}\cdot\mathbf{u}(k)$$

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{L}\cdot\left[\mathbf{y}(k+1) - \mathbf{C}\cdot\hat{\mathbf{x}}(k+1|k)\right]$$

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k+1) &= \mathbf{A}\cdot\hat{\mathbf{x}}(k|k) + \mathbf{B}\cdot\mathbf{u}(k) + \mathbf{L}\cdot\left[\mathbf{y}(k+1) - \mathbf{C}\cdot\hat{\mathbf{x}}(k+1|k)\right] \\ &= \mathbf{A}\cdot\hat{\mathbf{x}}(k|k) + \mathbf{B}\cdot\mathbf{u}(k) + \mathbf{L}\cdot\mathbf{y}(k+1) - \mathbf{LC}\cdot\hat{\mathbf{x}}(k+1|k) \\ &= (\mathbf{I}-\mathbf{LC})\mathbf{A}\cdot\hat{\mathbf{x}}(k|k) + (\mathbf{I}-\mathbf{LC})\mathbf{B}\cdot\mathbf{u}(k) + \mathbf{L}\cdot\mathbf{y}(k+1)\end{aligned}$$

Define estimation error to be $\boxed{\tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k|k)}$

$$\mathbf{x}(k+1) = \mathbf{A}\cdot\mathbf{x}(k) + \cancel{\mathbf{B}\cdot\mathbf{u}(k)} \qquad y(k) = C\cdot\mathbf{x}(k)$$

$$-\quad \hat{\mathbf{x}}(k+1|k+1) = (\mathbf{I}-\mathbf{LC})\mathbf{A}\cdot\hat{\mathbf{x}}(k|k) + \cancel{(\mathbf{I}-\mathbf{LC})\mathbf{B}\cdot\mathbf{u}(k)} + \mathbf{LC}\cdot(\mathbf{Ax}(k) + \cancel{\mathbf{B}\cdot\mathbf{u}(k)})$$

$$\tilde{\mathbf{x}}(k+1) = \underbrace{(\mathbf{I}-\mathbf{LC})\mathbf{A}\cdot\tilde{\mathbf{x}}(k)}_{}, \quad \tilde{\mathbf{x}}(0) = \mathbf{x}(0)$$

```
K = place(A', A'*C', P);
```

POLE PLACEMENT FOR P-C OBSERVER

- If (A,C) is observable, will $(A,CA) \equiv (A,\bar{C})$ remain to be observable?

$$Rank\begin{bmatrix} \bar{\mathbf{C}} \\ \bar{\mathbf{C}}\mathbf{A} \\ \vdots \\ \bar{\mathbf{C}}\mathbf{A}^{n-1} \end{bmatrix} = Rank\begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^n \end{bmatrix} = Rank\left\{\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \cdot \mathbf{A}\right\}(?=)n$$

The answer is yes, if the original system $(\mathbf{A},\mathbf{C})$ is observable and the system matrix $\mathbf{A}$ is nonsingular. Note the system matrix is always nonsingular if the discrete-time model is obtained from a continuous-time system, i.e. $\mathbf{A} = e^{\mathbf{F}T}$, where T is the sampling period.

REDUCED ORDER OBSERVER

- When the output is also an state, the observer can be of reduced order, to reduce complexity of the implementation. In general, the output is related to states in the form *y=Cx.*

Constrain the observed states $\hat{\mathbf{x}}(k|k)$ to satisfy $\boxed{\mathbf{y}(k) = \mathbf{C}\cdot\hat{\mathbf{x}}(k|k)}$

$$\begin{aligned}\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k|k) &= \mathbf{C}\cdot\tilde{\mathbf{x}}(k|k) = \mathbf{C}(\mathbf{I}-\mathbf{LC})\mathbf{A}\cdot\tilde{\mathbf{x}}(k-1|k-1)\\ &= (\mathbf{I}-\mathbf{CL})\mathbf{CA}\cdot\tilde{\mathbf{x}}(k-1|k-1) = \mathbf{0}\end{aligned}$$

L should be selected so that

$$\boxed{(\mathbf{I} - \underset{1\times n}{\mathbf{C}}\ \underset{n\times 1}{\mathrm{L}}) = \mathbf{0}}$$

EX8.8 REDUCED ORDER OBSERVER FOR DOUBLE INTEGRATOR

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \mathbf{x}(k) + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \cdot u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \mathbf{x}(k)$$

Observer gain $\mathbf{L}^T = \begin{bmatrix} l_1 & l_2 \end{bmatrix}$

To satisfy $(\mathbf{I} - \mathbf{CL}) = \mathbf{0} \quad \Longrightarrow \quad 1 - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} = 0 \quad \Rightarrow \quad l_1 = 1$

$$\hat{\mathbf{x}}(k+1|k+1) = (\mathbf{I} - \mathbf{LC})\mathbf{A} \cdot \hat{\mathbf{x}}(k|k) + (\mathbf{I} - \mathbf{LC})\mathbf{B} \cdot u(k) + \mathbf{L} \cdot y(k+1)$$

$$= \begin{bmatrix} 0 & 0 \\ -l_2 & 1 - l_2 T \end{bmatrix} \cdot \hat{\mathbf{x}}(k|k) + \begin{bmatrix} 0 \\ T - \dfrac{l_2 T^2}{2} \end{bmatrix} \cdot u(k) + \begin{bmatrix} 1 \\ l_2 \end{bmatrix} \cdot y(k+1)$$

$$\hat{x}_1(k+1|k+1) = y(k+1)$$

$$\hat{x}_2(k+1|k+1) = \left(1 - l_2 T\right) \hat{x}_2(k|k) + l_2 \left[y(k+1) - y(k) \right] + \left(T - \frac{l_2 T^2}{2} \right) u(k)$$

Ex8.8 (cont.)

$$\hat{x}_1(k+1|k+1) = y(k+1)$$

$$\hat{x}_2(k+1|k+1) = (1 - l_2 T)\hat{x}_2(k|k) + l_2[y(k+1) - y(k)] + \left(T - \frac{l_2 T^2}{2}\right)u(k)$$

The estimation of position is obtained directly from the Measurement.

$\boxed{l_2 = 0}$ Completely ignore measurement (no correction)

$$\hat{x}_2(k+1|k+1) = \hat{x}_2(k|k) + Tu(k)$$

$\boxed{l_2 = 1/T}$ Max gain (toss out history)

$$\hat{x}_2(k+1|k+1) = \frac{1}{T}[y(k+1) - y(k)] + \frac{T}{2}u(k)$$

In other word, the estimation of the position $\hat{x}_1$ is obtained directly from the measurement. l_2 can be selected to adjust the estimation algorithm for $\hat{x}_2$. For example, both $l_2 = 0$ or $l_2 = 1/T$ will result in two estimation algorithms that have physical interpretation. However, $l_2 > 1/T$ will result in negative poles and should be avoided. Hence, a reasonable design envelop for l_2 is between 0 and $1/T$.

OBSERVER-BASED COMPENSATOR

- Use observer in full-state feedback designs

<u>Continuous-Time Systems</u>	<u>Discrete-Time Systems</u>
$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{u}(t)$	$\mathbf{x}(k+1) = \mathbf{A} \cdot \mathbf{x}(k) + \mathbf{B} \cdot \mathbf{u}(k)$
$\mathbf{y}(t) = \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot \mathbf{u}(t)$	$\mathbf{y}(k) = \mathbf{C} \cdot \mathbf{x}(k) + \mathbf{D} \cdot \mathbf{u}(k)$

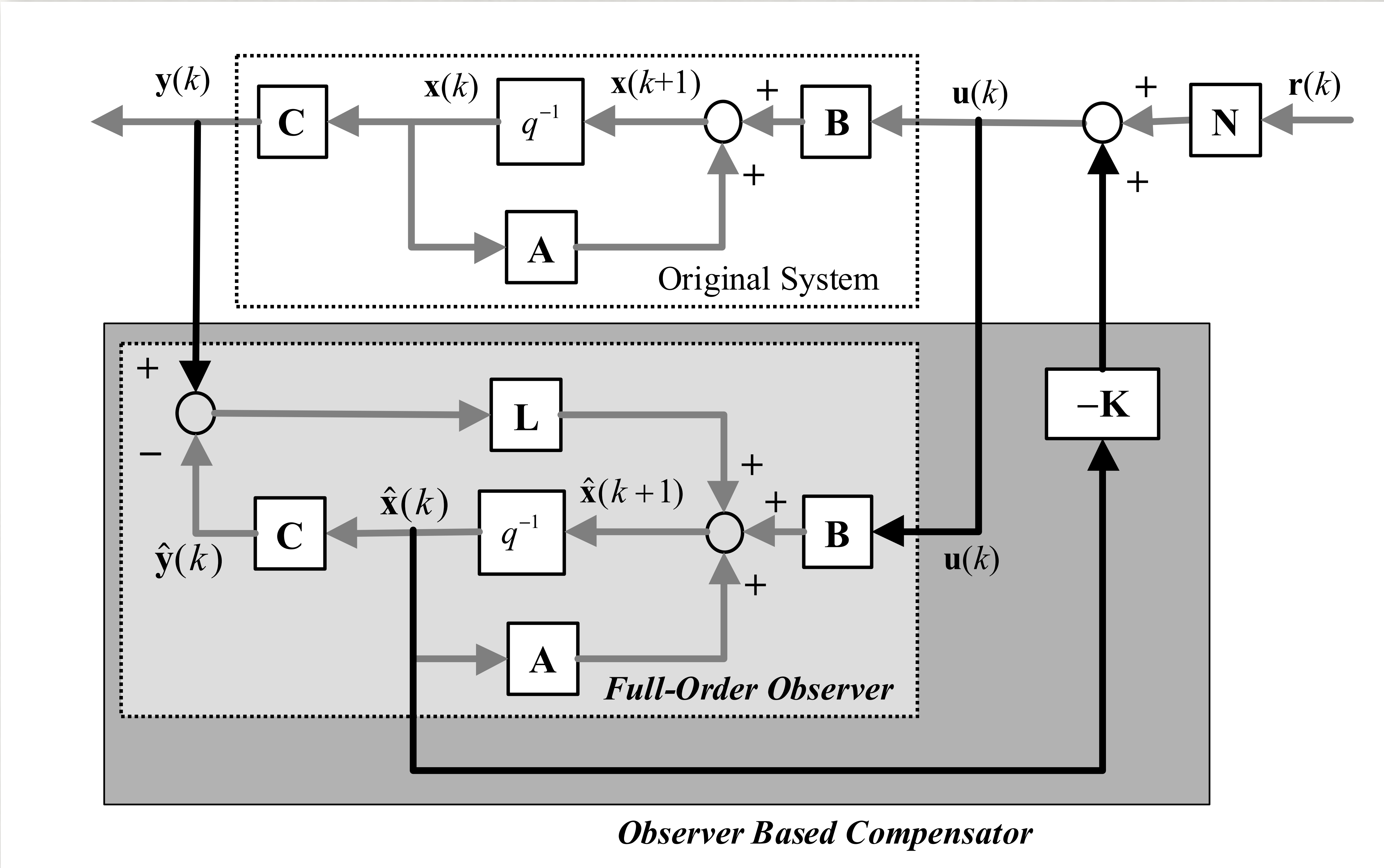
Observer:

$$\dot{\hat{\mathbf{x}}}(t) = (\mathbf{A} - \mathbf{LC}) \cdot \hat{\mathbf{x}}(t) + \mathbf{L} \cdot y(t) + \mathbf{B} \cdot u(t) \qquad \hat{\mathbf{x}}(k+1) = (\mathbf{A} - \mathbf{LC}) \cdot \hat{\mathbf{x}}(k) + \mathbf{L} \cdot y(k) + \mathbf{B} \cdot u(k)$$

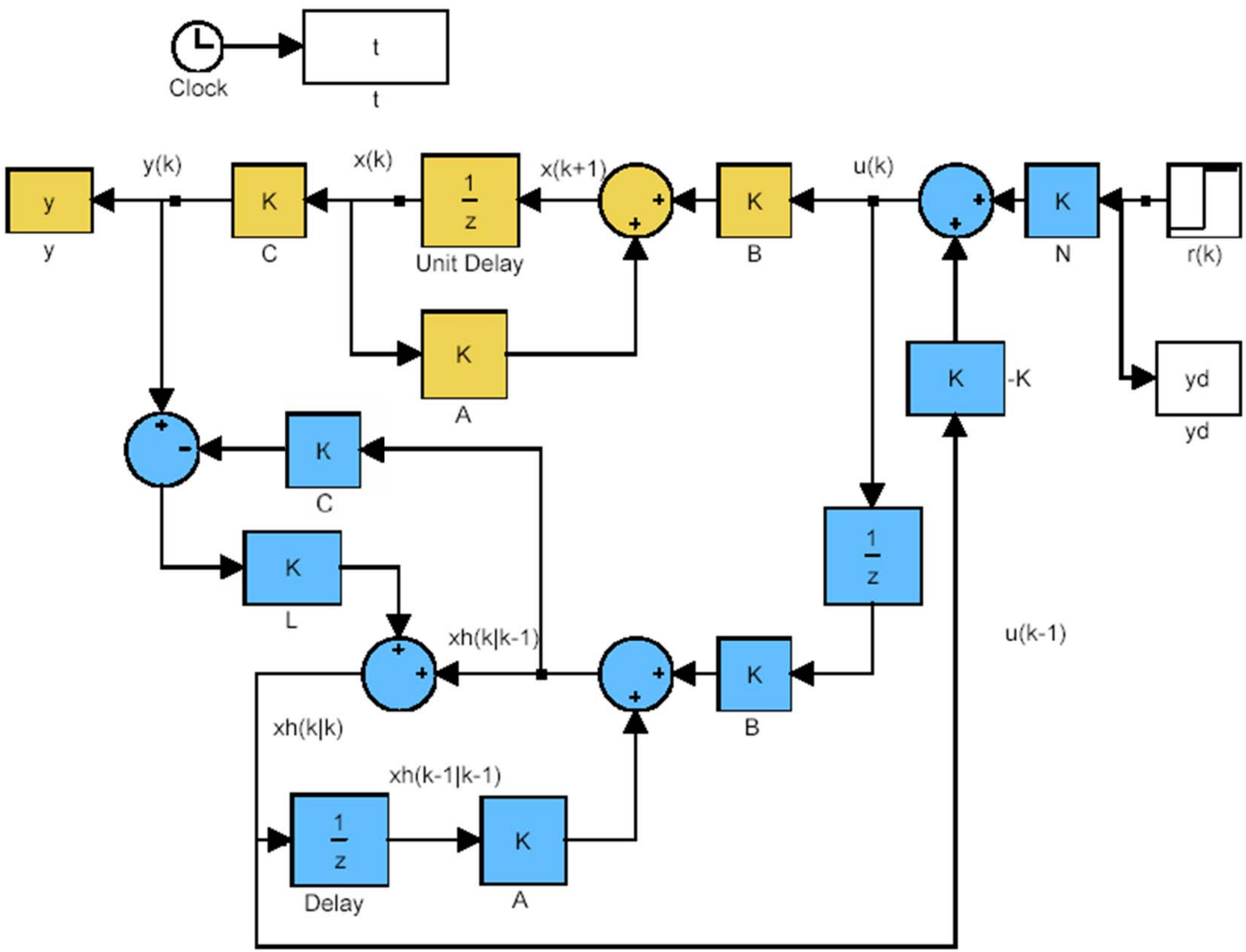
Controller

$$u(t) = -\mathbf{K} \cdot \hat{\mathbf{x}}(t) + \mathbf{N} \cdot r(t) \qquad u(k) = -\mathbf{K} \cdot \hat{\mathbf{x}}(k) + \mathbf{N} \cdot r(k)$$

OBSERVER-BASED COMPENSATOR (L)



Observer-Based Compensator (P-C)



CLOSED-LOOP SYSTEM (LUENBERGER OBSERVER)

$$\mathbf{x}(k+1) = \mathbf{A}\cdot\mathbf{x}(k) + \mathbf{B}\cdot\mathbf{u}(k) \qquad u(k) = -\mathbf{K}\cdot\hat{\mathbf{x}}(k) + \mathbf{N}\cdot r(k)$$

$$\hat{\mathbf{x}}(k+1) = \left(\mathbf{A} - \mathbf{LC}\right)\cdot\hat{\mathbf{x}}(k) + \mathbf{L}\cdot y(k) + \mathbf{B}\cdot u(k)$$

The *2nx1* system

$$\begin{bmatrix}\mathbf{x}(k+1)\\ \hat{\mathbf{x}}(k+1)\end{bmatrix} = \begin{bmatrix}\mathbf{A} & -\mathbf{BK}\\ \mathbf{LC} & \mathbf{A}-\mathbf{LC}-\mathbf{BK}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \hat{\mathbf{x}}(k)\end{bmatrix} + \begin{bmatrix}\mathbf{B}\\ \mathbf{B}\end{bmatrix}\cdot\left(\mathbf{N}\cdot r(t)\right)$$

$$y(k) = \begin{bmatrix}\mathbf{C} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \hat{\mathbf{x}}(k)\end{bmatrix} \qquad \tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$$

$$\begin{bmatrix}\mathbf{x}(k+1)\\ \tilde{\mathbf{x}}(k+1)\end{bmatrix} = \begin{bmatrix}\mathbf{A}-\mathbf{BK} & \mathbf{BK}\\ \mathbf{0} & \mathbf{A}-\mathbf{LC}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \tilde{\mathbf{x}}(k)\end{bmatrix} + \begin{bmatrix}\mathbf{B}\\ \mathbf{0}\end{bmatrix}\cdot\left(\mathbf{N}\cdot r(t)\right)$$

$$y(k) = \begin{bmatrix}\mathbf{C} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \tilde{\mathbf{x}}(k)\end{bmatrix}$$

CLOSED-LOOP SYSTEM (CONT.)

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \tilde{\mathbf{x}}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}-\mathbf{BK} & \mathbf{BK} \\ \mathbf{0} & \mathbf{A}-\mathbf{LC} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \tilde{\mathbf{x}}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \cdot \left(\mathbf{N} \cdot r(t)\right)$$

$$y(k) = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \tilde{\mathbf{x}}(k) \end{bmatrix}$$

poles:

$$\underbrace{\det \begin{bmatrix} z\mathbf{I}-\left(\mathbf{A}-\mathbf{BK}\right) & -\mathbf{BK} \\ \mathbf{0} & z\mathbf{I}-\left(\mathbf{A}-\mathbf{LC}\right) \end{bmatrix} = \underbrace{\det\left[z\mathbf{I}-\left(\mathbf{A}-\mathbf{BK}\right)\right]}_{\text{poles of FB control}} \cdot \underbrace{\det\left[z\mathbf{I}-\left(\mathbf{A}-\mathbf{LC}\right)\right]}_{\text{poles of observer}}}$$

Separation principle of
observer-based compensator

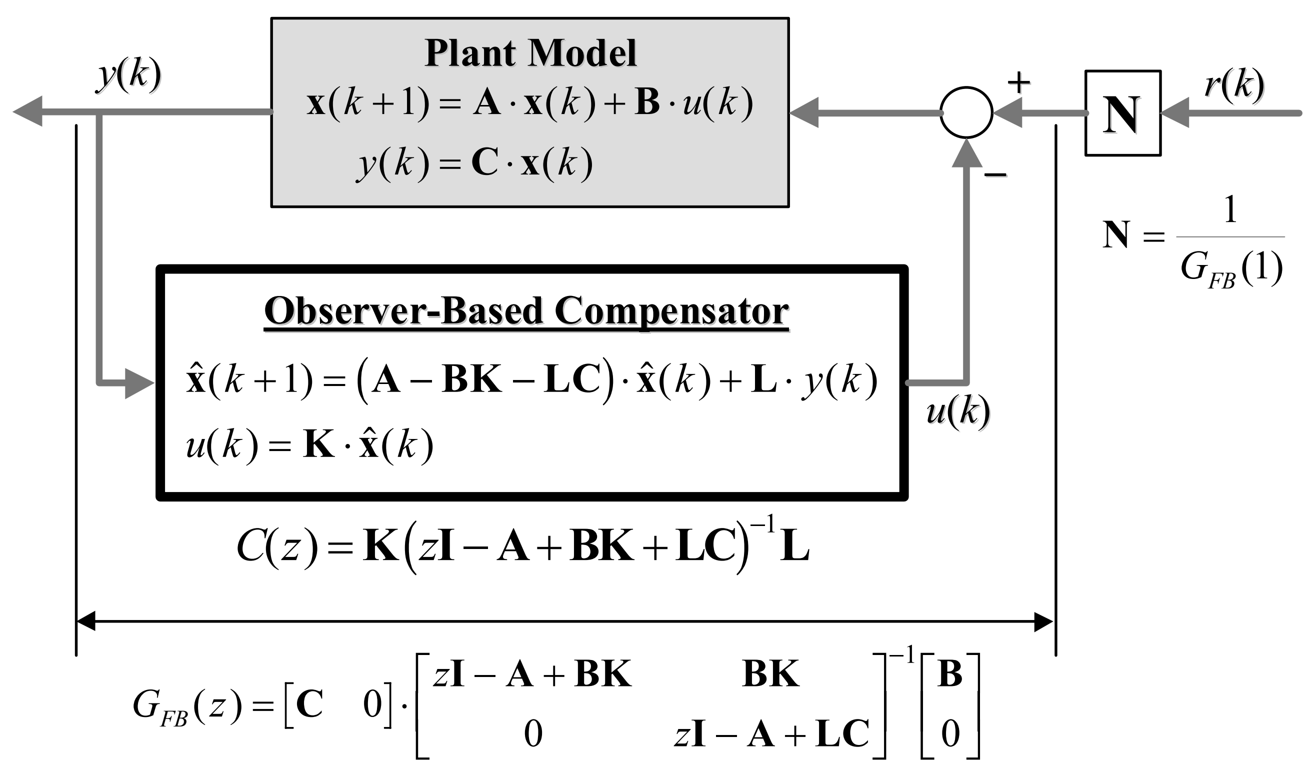
REFERENCE INPUT SCALING

- The pulse transfer function from reference input $r(k)$ to the measured output $y(k)$ is

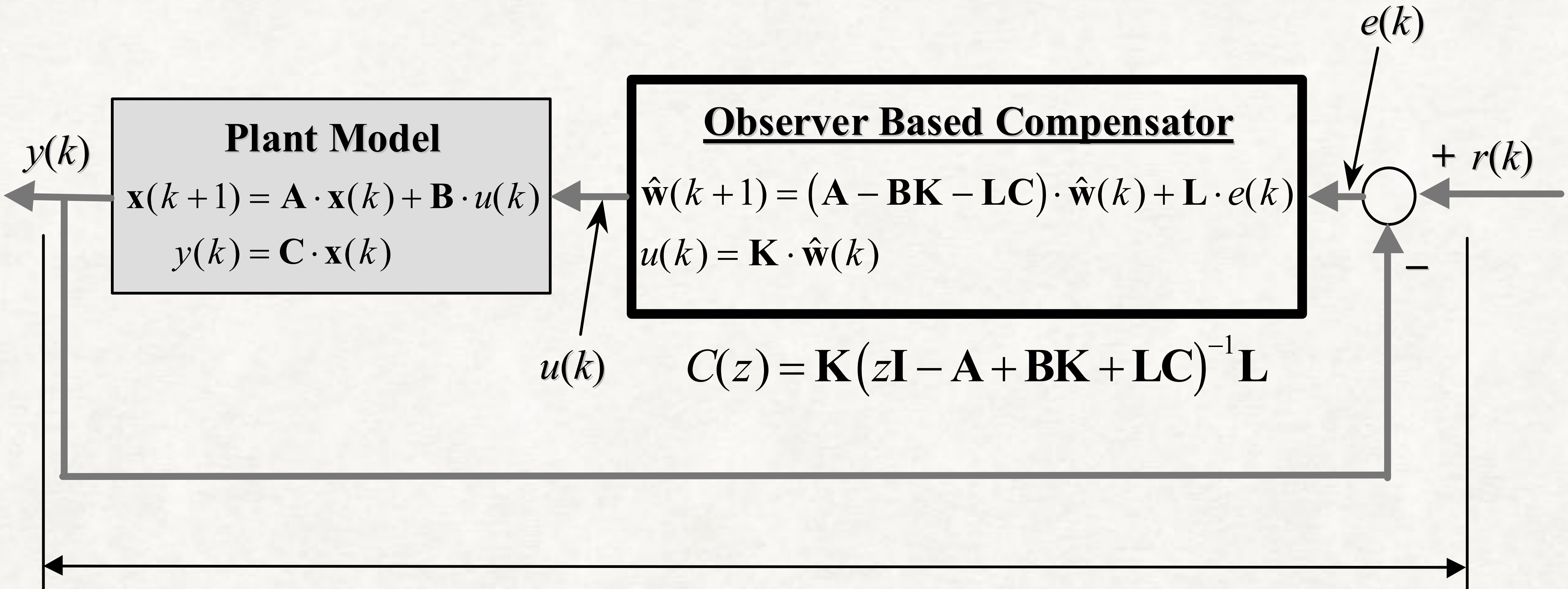
$$G_{CL}(z) = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} z\mathbf{I} - \mathbf{A} + \mathbf{BK} & \mathbf{BK} \\ \mathbf{0} & z\mathbf{I} - \mathbf{A} + \mathbf{LC} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{N}$$

- To maintain unit closed-loop steady-state gain

$$\mathbf{N} = \left\{ \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} - \mathbf{A} + \mathbf{BK} & \mathbf{BK} \\ \mathbf{0} & \mathbf{I} - \mathbf{A} + \mathbf{LC} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \right\}^{-1}$$



OBSERVER-BASED COMPENSATOR IN THE FORWARD PATH



$$G_{CL}(z) = \begin{bmatrix}\mathbf{C} & 0\end{bmatrix}\cdot\begin{bmatrix} z\mathbf{I} - \mathbf{A} & \mathbf{BK} \\ \mathbf{LC} & z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{LC}\end{bmatrix}^{-1}\begin{bmatrix}0 \\ \mathbf{L}\end{bmatrix}$$

$$\boxed{N = \frac{1}{G_{CL}(1)}}$$

Since closed-loop zeros will affect the transient performance of the system, placing the compensator in the forward path requires careful consideration and the designer should make sure that the additional zeros will not introduce undesirable transient response.

P-C OBSERVER-BASED COMPENSATOR

$$\mathbf{x}(k+1) = \mathbf{A}\cdot\mathbf{x}(k) + \mathbf{B}\cdot\mathbf{u}(k)$$

$$u(k) = -\mathbf{K}\cdot\hat{\mathbf{x}}(k\,|\,k) + \mathbf{N}\cdot r(k)$$

$$\tilde{\mathbf{x}}(k+1) = (\mathbf{I}-\mathbf{LC})\mathbf{A}\cdot\tilde{\mathbf{x}}(k), \quad \tilde{\mathbf{x}}(0) = \mathbf{x}(0)$$

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

$$\Rightarrow \begin{bmatrix} \mathbf{x}(k+1) \\ \tilde{\mathbf{x}}(k+1|k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}-\mathbf{BK} & \mathbf{BK} \\ \mathbf{0} & (\mathbf{I}-\mathbf{LC})\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \tilde{\mathbf{x}}(k|k) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}\cdot(\mathbf{N}\cdot r(t))$$

$$y(k) = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \tilde{\mathbf{x}}(k|k) \end{bmatrix}$$

$$\Rightarrow \det\begin{bmatrix} z\mathbf{I}-(\mathbf{A}-\mathbf{BK}) & -\mathbf{BK} \\ \mathbf{0} & z\mathbf{I}-(\mathbf{A}-\mathbf{LCA}) \end{bmatrix} = \underbrace{\underbrace{\det\left[z\mathbf{I}-(\mathbf{A}-\mathbf{BK})\right]}_{\text{poles of FB control}}\cdot\underbrace{\det\left[z\mathbf{I}-(\mathbf{A}-\mathbf{LCA})\right]}_{\text{poles of P-C observer}}}_{\text{Separation principle of P-C observer-based compensator}} = 0$$

poles of
FB control

poles of
P-C observer

Separation principle of
P-C observer-based compensator