CSE 590K: Analysis and Control of Computing Systems Using Linear Discrete-Time System Theory:

State-Space Models for LTI Systems

Joseph L. Hellerstein, Jie Liu
Microsoft

Feb. 4, 2008

Agenda

- State space Modeling
  - States & state variables
  - State space models
  - State space and Transfer Function formats
  - Realization Theory
- Model Analysis
  - State trajectories
  - Stability
  - Steady state properties
  - Transient properties
- Controllability & Observability
  - State feedback control
  - Observers

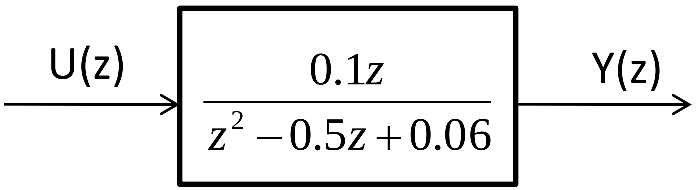
Motivation: why use State Space models?

- High dimension systems
  - Writing difference equations can become tedious
- MIMO (Multi-input, Multi-output) systems
- Realization ambiguity

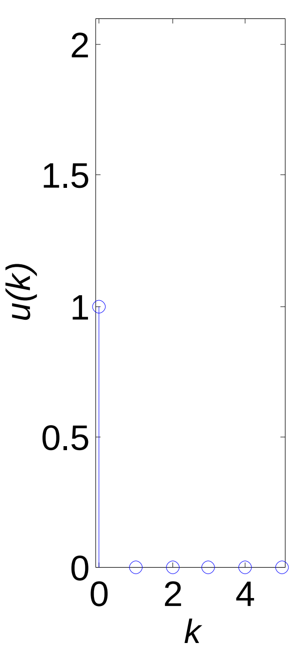
- Vector-Matrix notation provides a uniform, compact representation
- Easier to manipulate and represent
- Many concepts from transfer function analysis can be applied

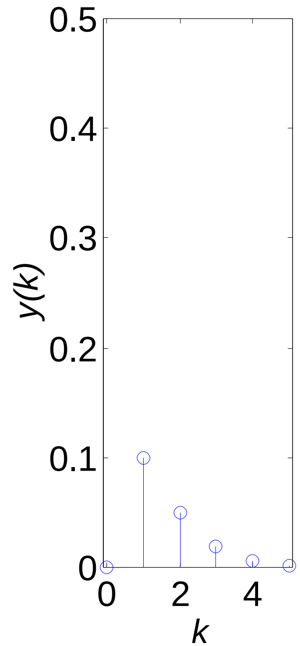
Realization of Transfer Functions

Recall from last lecture: Decompose into a sum of geometrics



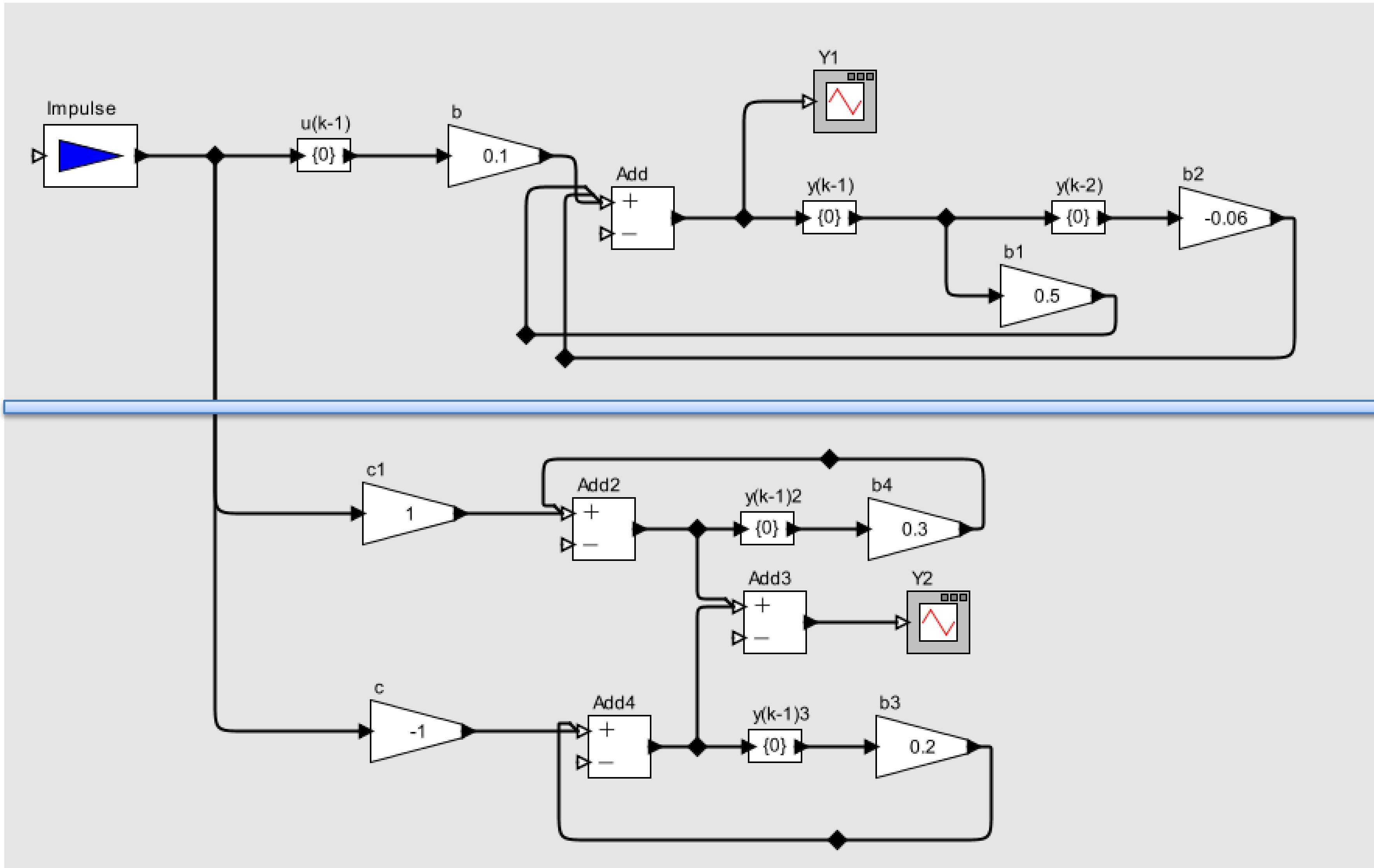
$$G(z)=\frac{0.1z}{z^2-0.5z+0.06}$$
$$=\frac{z}{z-0.3}-\frac{z}{z-0.2}$$
$$=1+0.3z^{-1}+0.09z^{-2}+\ldots-1-0.2z^{-1}-0.04z^{-2}+\ldots$$
$$=0.1z^{-1}+0.05z^{-2}+\ldots$$



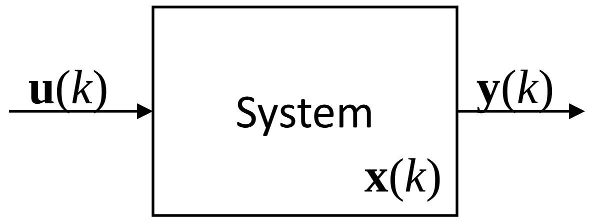


- **<u>Partial fraction expansion</u>** allows rational polynomials to be decomposed into a sum of geometrics
- Poles of the original polynomial are the poles of the geometrics

Realizations of TF



State Variables

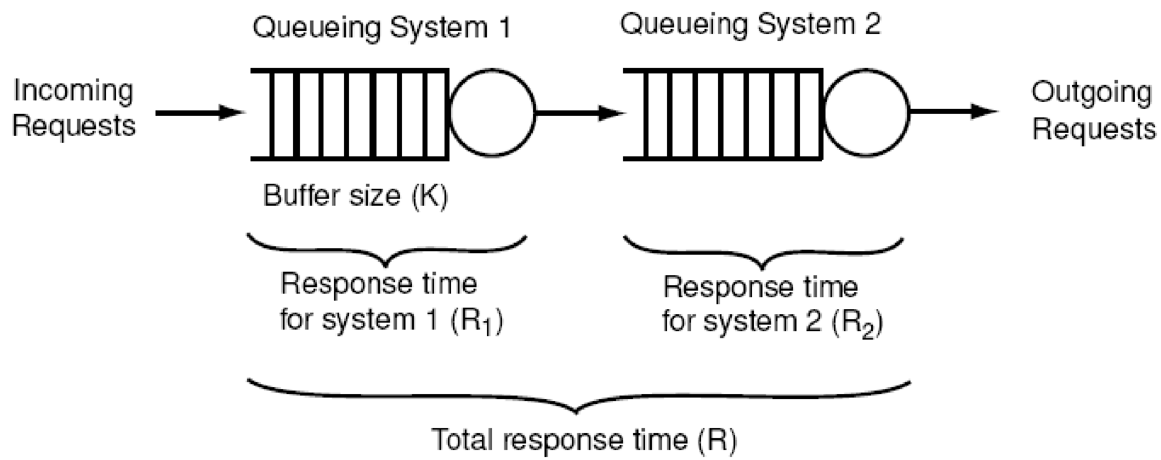


- Variables that *uniquely* define the evolution of the system
- Used to express system dynamics
  - May not even be measurable
- Similar to State in state automata, MDPs, etc
- Typically, multiple state variables will be used
  - i.e., state ≡ *n*-tuple <$x_1, x_2, \ldots, x_n$>
  - represented as a vector (boldface)

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}$$

- State variables need *NOT* be completely independent or unrelated
  - Common to have "history" terms: $x_2 = x_1(k\text{-}1)$, etc

State Variables Example: Tandem Queue

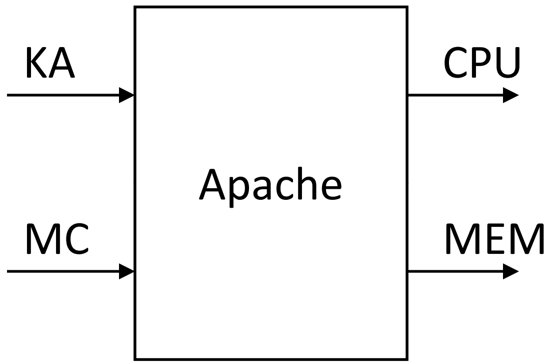


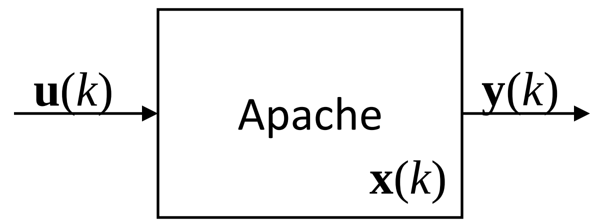
- Control Input: buffer size at Queue 1 (*K*)
- Output: End-to-end response time (*R*)

- Idea: Model each queue separately
- State variables: R_1, R_2

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} R_1(k) - \overline{R_1} \\ R_2(k) - \overline{R_2} \end{bmatrix}$$

State Variables Example: Apache MIMO system





- MIMO System
- Control Inputs: KeepAlive (KA), MaxClients (MC)
- System Outputs: CPU, MEMory utilization

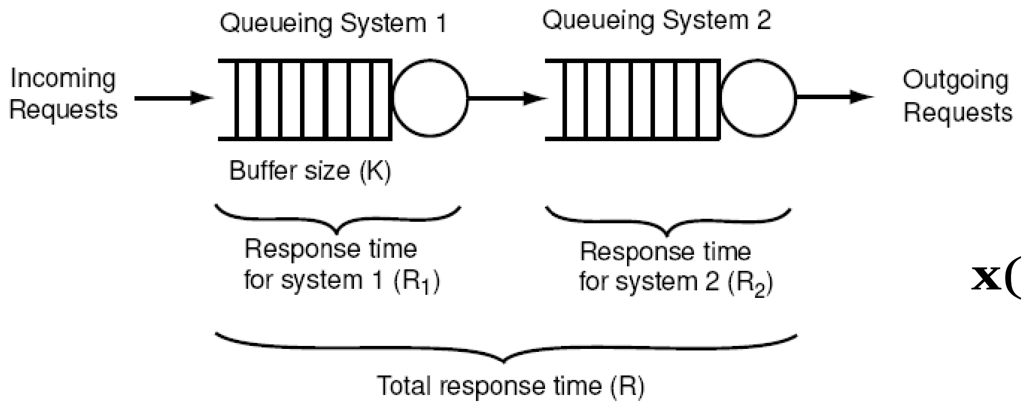
- State variables:
  - –Current & Past values of CPU, MEM

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} = \begin{bmatrix} \text{CPU}(k) - \overline{\text{CPU}} \\ \text{MEM}(k) - \overline{\text{MEM}} \\ \text{CPU}(k-1) - \overline{\text{CPU}} \\ \text{MEM}(k-1) - \overline{\text{MEM}} \end{bmatrix}$$

State Space Models

- Canonical Representation consists of two parts
  1. State evolution dynamics
     $$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$$
  2. How state determines the system output
     $$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k))$$
  – <u>NOTE</u>: **x**(k) encapsulates *entire* history required for the dynamics
    - Do not need **x**(k-1), **x**(k-2), etc.

State Space Model: Example



$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} R_1(k) - \overline{R_1} \\ R_2(k) - \overline{R_2} \end{bmatrix}$$

$$\begin{aligned} x_1(k+1) &= a_{11}x_1(k) + bu(k) \\ x_2(k+1) &= a_{22}x_2(k) + a_{21}x_1(k) \\ \\ y(k) &= x_1(k) + x_2(k) \end{aligned}$$

$\longrightarrow$

$$\mathbf{x}(k+1) = \begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u(k)$$

$$\mathbf{y}(k) = [1 \quad 1] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

LTI State Space Model: General Form

$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right)$$
$$= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

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

Tandem Queue Example:

$$\mathbf{x}(k+1) = \begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u(k)$$

$$\mathbf{y}(k) = [1 \quad 1] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

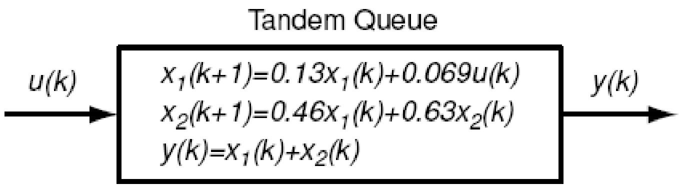
$\longrightarrow$

$$\mathbf{A} = \begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix}$$

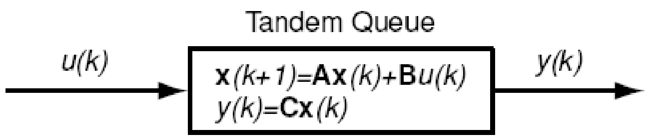
$$\mathbf{B} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

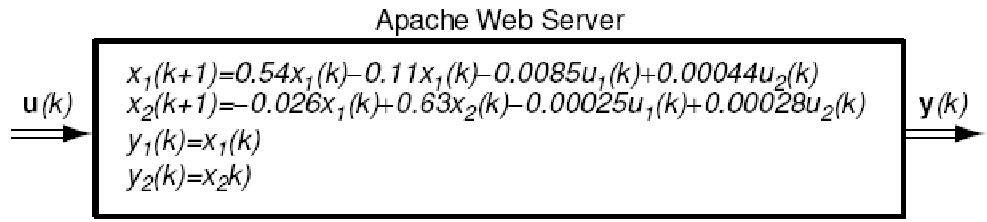
Comparing Scalar vs Matrix Descriptions



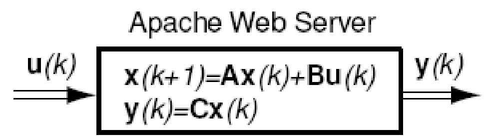
(a) Scalar description



(b) Matrix description



(a) Scalar description



(b) Matrix description

- Advantages
  - Compact notation
  - Uniform format
  - Analysis based on properties of **A**, **B**, and **C**

Realization Theory

- There are many ways to convert difference equations to state space representations.
- The ones with minimum number of z^{-1} is called minimum realizations.
  - Control canonical form
  - Observer canonical form
  - Jordan canonical form

Example: Converting ARX models to State Space

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + \cdots + a_n y(k-n) + b_1 u(k-1)$$

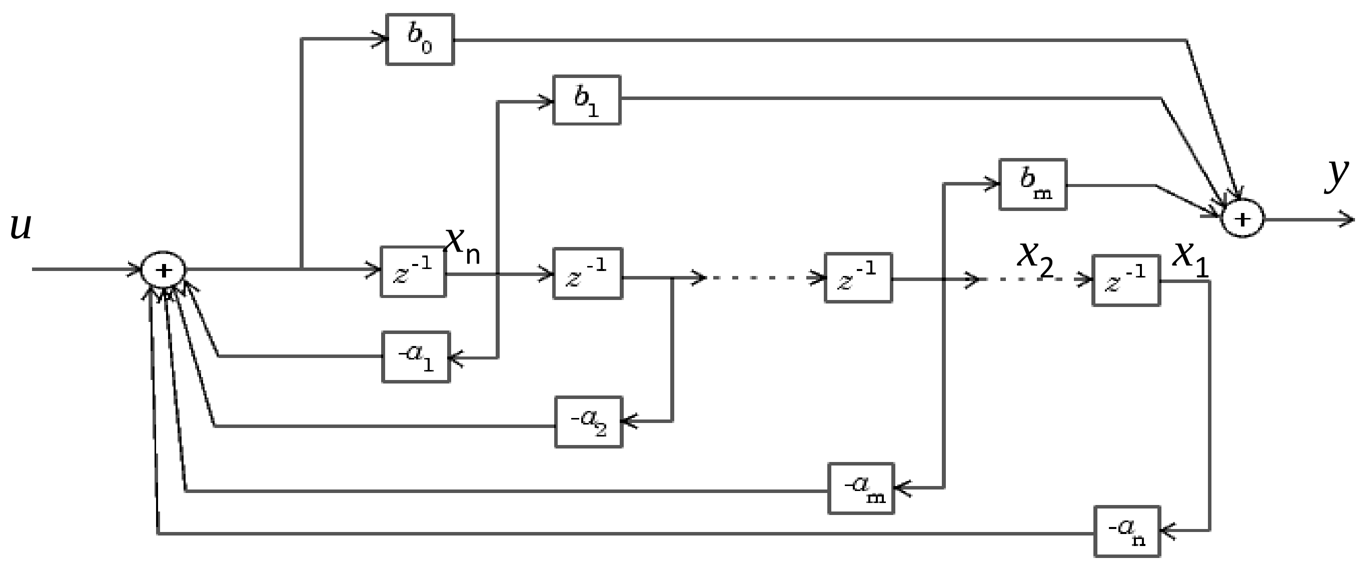
$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} = \begin{bmatrix} y(k-n+1) \\ y(k-n+2) \\ \vdots \\ y(k-1) \\ y(k) \end{bmatrix}$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ a_n & a_{n-1} & \cdots & a_2 & a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b_1 \end{bmatrix} u(k)$$

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

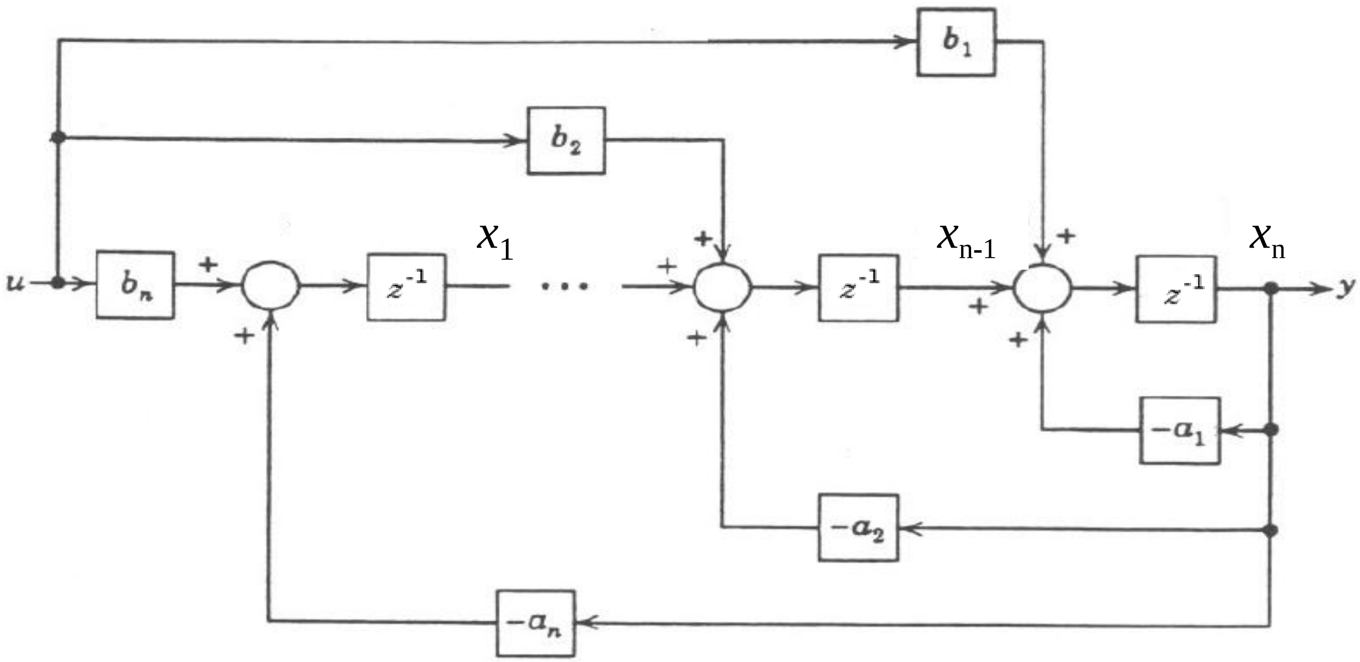
Controller Canonical Realization

$$\frac{Y(z)}{U(z)} = (b_0 + b_1 z^{-1} + \dots + b_m z^{-m}) \quad \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}$$



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 \\ -a_n & -a_{n-1} & \dots & -a_2 & -a_1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad C = [0, \dots, b_m, b_{m-1}, \cdots, b_1]$$

Observer Canonical Realization

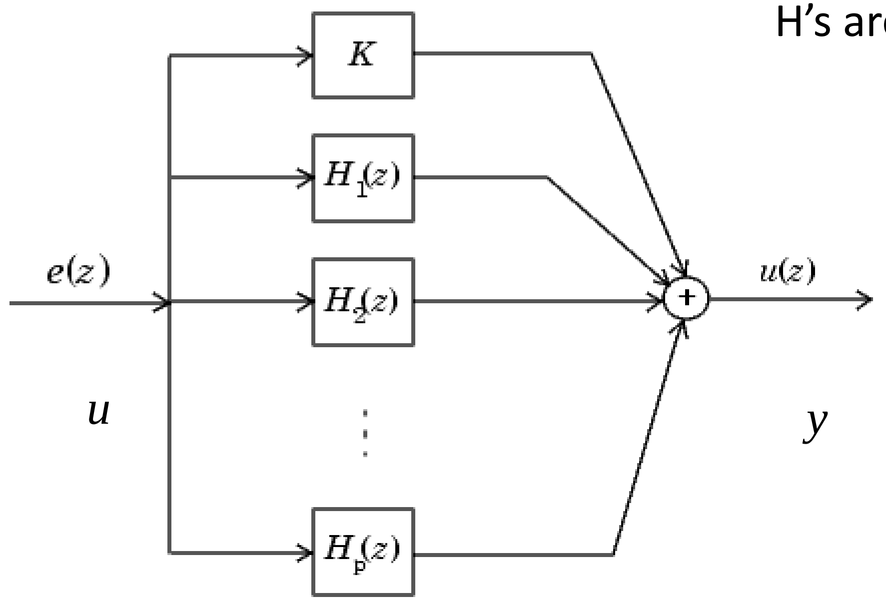


$$A = \begin{bmatrix} 0 & 0 & \dots & 0 & -a_n \\ 1 & 0 & \dots & 0 & -a_{n-1} \\ 0 & 1 & 0 & \dots & -a_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -a_1 \end{bmatrix} \quad B = \begin{bmatrix} b_n \\ b_{n-1} \\ b_{n-2} \\ \vdots \\ b_1 \end{bmatrix} \quad C = [0, \cdots 0, 1]$$

Jordan Canonical Realization

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_m z^{-m}}{1 + a_1 z^{-1} + \ldots + a_n z^{-n}} = K + H_1(z) + H_2(z) + \ldots + H_p(z)$$

H's are first or second order systems.



What are the A, B, C, D?

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_p \end{bmatrix} \quad B = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_p \end{bmatrix} \quad C = [C_1, C_2, \ldots C_p] \quad D = K$$

State Space Analysis

- State trajectories
- Stability
- Poles

Linear Algebra Review

- Matrix
  - rows, columns, left*, right*
  - Identity matrix, I
  - transpose: A^T
  - complex conjugate: A*
  - determinant
  - rank: The number of linearly independent rows (columns)

  (how to check full rank?)
  - singular matrix
  - inverse (if not singular):
    - $AA^{-1}=I$, $A^{-1}A=I$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

$$A^* = A^T = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 4 & 0 \\ 3 & 5 & 6 \end{bmatrix}$$

$$\det(A) = 22$$

$$A^{-1} = \begin{bmatrix} 12/11 & -6/11 & -1/11 \\ 5/22 & 3/22 & -5/22 \\ -2/11 & 1/11 & 2/11 \end{bmatrix}$$

Eigenvalues

- For a square matrix A, there exist scalars λ and vectors v, such that

$$Av = \lambda v, \text{ or } (\lambda I - A)v = 0$$

  - λ is called an eigenvalue of A and v is the corresponding eigenvector.
  - If A is full rank, then there are n eigenvalues and n linearly independent eigenvectors.
- Eigendecomposition:

$$A = P\Lambda P^{-1}$$

State Trajectories

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

$$\begin{aligned}
\mathbf{x}(1) &= \mathbf{A}\mathbf{x}(0) + \mathbf{B}\mathbf{u}(0) \\
\mathbf{x}(2) &= \mathbf{A}\mathbf{x}(1) + \mathbf{B}\mathbf{u}(1) \\
&= \mathbf{A}^2\mathbf{x}(0) + \mathbf{A}\mathbf{B}\mathbf{u}(0) + \mathbf{B}\mathbf{u}(1) \\
\mathbf{x}(3) &= A\mathbf{x}(2) + \mathbf{B}\mathbf{u}(2) \\
&= \mathbf{A}^3\mathbf{x}(0) + \mathbf{A}^2\mathbf{B}\mathbf{u}(0) + \mathbf{A}\mathbf{B}\mathbf{u}(1) + \mathbf{B}\mathbf{u}(2)
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}(k) &= \mathbf{A}^k\mathbf{x}(0) + \mathbf{A}^{k-1}\mathbf{B}\mathbf{u}(0) + \mathbf{A}^{k-2}\mathbf{B}\mathbf{u}(1) + \cdots + \mathbf{A}\mathbf{B}\mathbf{u}(k-2) + \mathbf{B}\mathbf{u}(k-1) \\
&= \mathbf{A}^k\mathbf{x}(0) + \sum_{i=0}^{k-1} \mathbf{A}^{k-1-i}\mathbf{B}\mathbf{u}(i)
\end{aligned}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) = \mathbf{C}\mathbf{A}^k\mathbf{x}(0) + \mathbf{C}\sum_{i=0}^{k-1} \mathbf{A}^{k-1-i}\mathbf{B}\mathbf{u}(i)$$

Vector/Matrix Z-Transform

- Straightforward extension of the scalar case:

$$\mathbf{X}(z) = Z[\mathbf{x}(k)] = \begin{bmatrix} Z[x_1(k)] \\ Z[x_2(k)] \\ \vdots \\ Z[x_n(k)] \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{\infty} x_1(i) z^{-i} \\ \sum_{i=0}^{\infty} x_2(i) z^{-i} \\ \vdots \\ \sum_{i=0}^{\infty} x_n(i) z^{-i} \end{bmatrix}$$

Solving State Space Equations using Z-Transform

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

Take the Z-transform:

$$z\mathbf{X}(z) - z\mathbf{x}(0) = \mathbf{A}\mathbf{X}(z) + \mathbf{B}\mathbf{U}(z)$$

Collect terms:

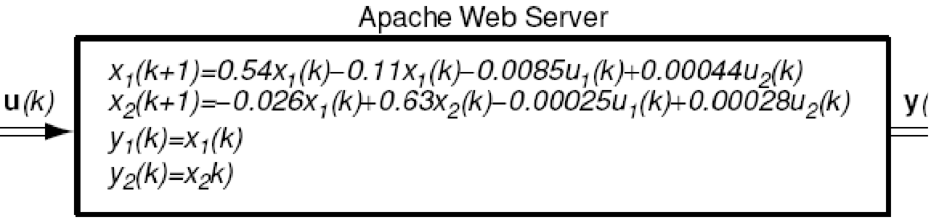
$$(z\mathbf{I} - \mathbf{A})\mathbf{X}(z) = z\mathbf{x}(0) + \mathbf{B}\mathbf{U}(z)$$
$$\mathbf{X}(z) = (z\mathbf{I} - \mathbf{A})^{-1}(z\mathbf{x}(0) + \mathbf{B}\mathbf{U}(z))$$

Output equation:

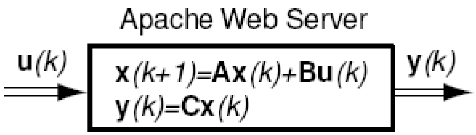
$$\mathbf{Y}(z) = \mathbf{C}\mathbf{X}(z)$$
$$= \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}(z\mathbf{x}(0) + \mathbf{B}\mathbf{U}(z))$$

Final Step: Invert the (column of) Z-transforms

Exercise: Solving State Space Equations

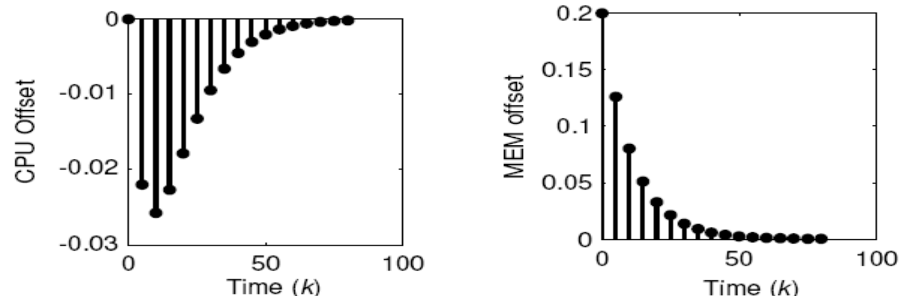


(a) Scalar description



(b) Matrix description

$$\mathbf{A} = \begin{bmatrix} 0.54 & -0.11 \\ -0.026 & 0.63 \end{bmatrix}$$



(a) Transient response of CPU (b) Transient response of MEM

$$x(0) = \begin{bmatrix} 0 \\ 0.2 \end{bmatrix}, \mathbf{U}(z) = 0$$

$$\mathbf{Y}(z) = \mathbf{C}(z\mathbf{I}-\mathbf{A})^{-1}(zx(0)+\mathbf{B}\mathbf{U}(z))$$

Hint: $\mathbf{A},\mathbf{B},\mathbf{C} = ?$

$(z\mathbf{I}-\mathbf{A})^{-1} = ?$

$\mathbf{Y}(z) = ?$

$\mathbf{y}(k) = ?$

$$(z\mathbf{I}-\mathbf{A})^{-1} = \frac{1}{\det(z\mathbf{I}-\mathbf{A})}\begin{bmatrix} z-0.63 & 0.11 \\ -0.26 & z-0.54 \end{bmatrix},$$

where

$$\det(z\mathbf{I}-\mathbf{A}) = (z-0.54)(z-0.63)-(0.11)-(0.26)$$

$$\mathbf{Y}(z) = \begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} \frac{-0.022z}{z^2-1.17z+0.34} \\ \frac{0.2z-1.08}{z^2-1.17z+0.34} \end{bmatrix}$$

$$\begin{aligned} y_1(k) &= (0.125)(0.54)^{k-1} - (0.147)(0.63)^{k-1} \\ y_2(k) &= (0.2)(0.63)^k \end{aligned}$$

State Space Transfer Function Matrix

System Model:

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

Z-transform:

$$\mathbf{Y}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}(zx(0) + \mathbf{B}\mathbf{U}(z))$$

If $x(0)=0$, then:

$$\mathbf{Y}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(z)$$

Define transfer function matrix:

$$\mathbf{G}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$$

So that:

$$\mathbf{Y}(z) = \mathbf{G}(z)\mathbf{U}(z)$$

NOTE: g_{ij} represents effect of j^{th} input on i^{th} output

Poles and Stability

System Model:

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

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

Z-transform:

$$\mathbf{Y}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}(zx(0) + \mathbf{B}\mathbf{U}(z))$$

Transfer Function: $\mathbf{G}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$

Characteristic Equation: $\det(z\mathbf{I} - \mathbf{A}) = 0$

Characteristic Polynomial: → $\det(z\mathbf{I} - \mathbf{A})$

Poles of G(z) are solutions of Characteristic Equation

(aka. *Eigenvalues* of $\mathbf{A}$)

<u>BIBO Stability</u>: All poles are inside unit circle [same as SISO systems]

Example: Apache Model

$$\mathbf{A} = \begin{bmatrix} 0.54 & -0.11 \\ -0.026 & 0.63 \end{bmatrix}$$

$$z\mathbf{I} - \mathbf{A} = \begin{bmatrix} z - 0.54 & 0.11 \\ 0.026 & z - 0.63 \end{bmatrix}$$

$$\begin{aligned} \det(z\mathbf{I} - \mathbf{A}) &= (z - 0.54)(z - 0.63) - (0.11) - (0.26) \\ &= z^2 - 1.17z + 0.337 = (z - 0.65)(z - 0.52) \end{aligned}$$

$\Rightarrow$Poles at: 0.65, 0.52
$\Rightarrow$Open-loop stable

Steady-State Gain

Recall: Final-value theorem: $y_{SS} = \lim_{z \to 1} (z-1) Y(z)$

Unit Step: $u(k) = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \mathbf{U}(z) = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \frac{z}{z-1}$

$$
\begin{aligned}
\mathbf{y}_{ss} &= \lim_{z \to 1} (z-1)\mathbf{Y}(z) \\
&= \lim_{z \to 1} (z-1)\mathbf{C}(z\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}\mathbf{U}(z) \\
&= \lim_{z \to 1} (z-1)\mathbf{C}(z\mathbf{I}-\mathbf{A})^{-1} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \frac{z}{z-1} \\
&= \mathbf{C}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{B} \\
&= \mathbf{G}(1)
\end{aligned}
$$

$$\boxed{\text{Steady state gain} = \mathbf{G}(1) = \mathbf{C}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}}$$

Controllability

- Idea: can we drive the system into *any* (achievable) state?

$$\mathbf{x}(n) = \mathbf{A}^{n-1}\mathbf{B}u(0) + \mathbf{A}^{n-2}\mathbf{B}u(1) + \cdots + \mathbf{B}u(n-1)$$

[Assume $x(0)=0$]

$$= \begin{bmatrix} \mathbf{A}^{n-1}\mathbf{B} & \mathbf{A}^{n-2}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(n-1) \end{bmatrix}$$

$$= \mathcal{C} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(n-1) \end{bmatrix}$$

$$\mathcal{C}^{-1}\mathbf{x}(n) = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(n-1) \end{bmatrix}$$

<u>Controllability matrix</u>:

$$\mathcal{C} = \begin{bmatrix} \mathbf{A}^{n-1}\mathbf{B} & \mathbf{A}^{n-2}\mathbf{B} & \cdots & \mathbf{AB} & \mathbf{B} \end{bmatrix}$$

Controllability $\Leftrightarrow$ ***C*** is invertible

Example

Notes with sensor:

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.43 & 0 \\ -0.037 & 0.64 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.47 \\ 0.080 \end{bmatrix} u(k)$$

1. Determine Controllability.
2. Can drive the system to x=[1 1]' ?

$$\mathbf{C} = \begin{bmatrix} 0.47 & 0.2 \\ 0.08 & 0.03 \end{bmatrix} \qquad \mathbf{C}^{-1} = \frac{-1}{0.0019} \begin{bmatrix} 0.03 & -0.2 \\ -0.08 & 0.47 \end{bmatrix}$$

$$\begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} = \mathbf{C}^{-1} x = \frac{-1}{0.0019} \begin{bmatrix} 0.03 & -0.2 \\ -0.08 & 0.47 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 89.5 \\ -205.3 \end{bmatrix}$$

Controller Canonical Form

- Controller canonical system is controllable.

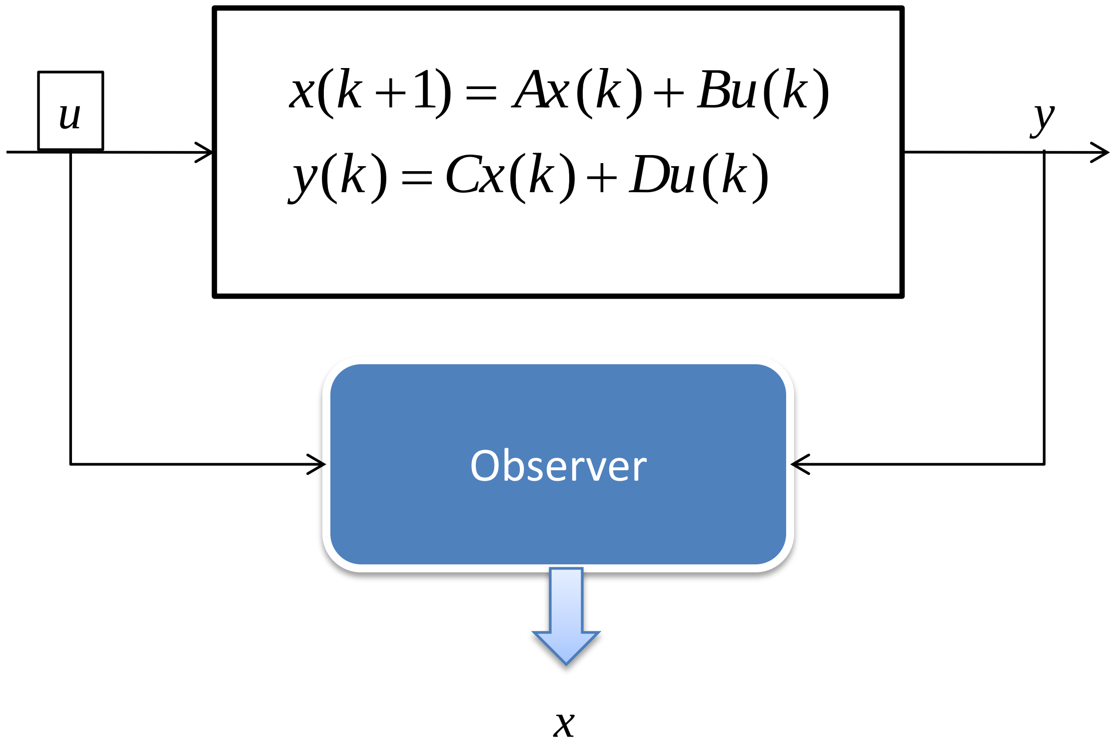
verify: $$A=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 \\ -a_n & -a_{n-1} & \ldots & -a_2 & -a_1 \end{bmatrix} \quad B=\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad C=[0,\ldots,b_m,b_{m-1},\cdots,b_1]$$

- All controllable system can be transformed into the controller canonical form
- Control by state feedback $u=-k_n x_1 - k_{n-1}x_2 - \ldots - k_1 x_n - v$

$$x(k+1)=(A-bk)x(k)+bv(k)=\begin{bmatrix} 0 & 1 & \ldots & 0 \\ 0 & 0 & \ddots & \\ 0 & \ldots & \ddots & 1 \\ -a_n-k_n & -a_{n-1}-k_{n-1} & & -a_1-k_1 \end{bmatrix} x(k)+bv(k)$$

Recall: poles are defined by $\det(zI-A)=0$

- But, how to obtain states if they are not measurable?



Observability

- Idea: can we infer all states of system from observing the output?
  - Eg, consider Tandem Queue

Observability matrix:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C}\mathbf{A}^{n-1} \\ \mathbf{C}\mathbf{A}^{n-2} \\ \vdots \\ \mathbf{C}\mathbf{A} \\ \mathbf{C} \end{bmatrix}$$

Observability $\Leftrightarrow$ *O* is invertible

$$\mathbf{x}(k) = \mathbf{A}^k\mathbf{x}(0) + \sum_{i=0}^{k-1} \mathbf{A}^{k-1-i}\mathbf{B}\mathbf{u}(i)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) = \mathbf{C}\mathbf{A}^k\mathbf{x}(0) + \mathbf{C}\sum_{i=0}^{k-1} \mathbf{A}^{k-1-i}\mathbf{B}\mathbf{u}(i)$$

$\mathbf{u} = 0$ implies that

$$y(k) = \mathbf{C}\mathbf{A}^k\mathbf{x}(0)$$

$$\begin{bmatrix} y(n-1) \\ y(n-2) \\ \vdots \\ y(1) \\ y(0) \end{bmatrix} = \begin{bmatrix} \mathbf{C}\mathbf{A}^{n-1}\mathbf{x}(0) \\ \mathbf{C}\mathbf{A}^{n-2}\mathbf{x}(0) \\ \vdots \\ \mathbf{C}\mathbf{A}\mathbf{x}(0) \\ \mathbf{C}\mathbf{x}(0) \end{bmatrix} = \begin{bmatrix} \mathbf{C}\mathbf{A}^{n-1} \\ \mathbf{C}\mathbf{A}^{n-2} \\ \vdots \\ \mathbf{C}\mathbf{A} \\ \mathbf{C} \end{bmatrix}\mathbf{x}(0)$$

Example

Tandem Queue:

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.13 & 0 \\ 0.46 & 0.63 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.069 \\ 0 \end{bmatrix}$$

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

1. Determine Observability
2. What initial conditions when y(0)=1.22, y(1)=2 ?

Observer Canonical Form

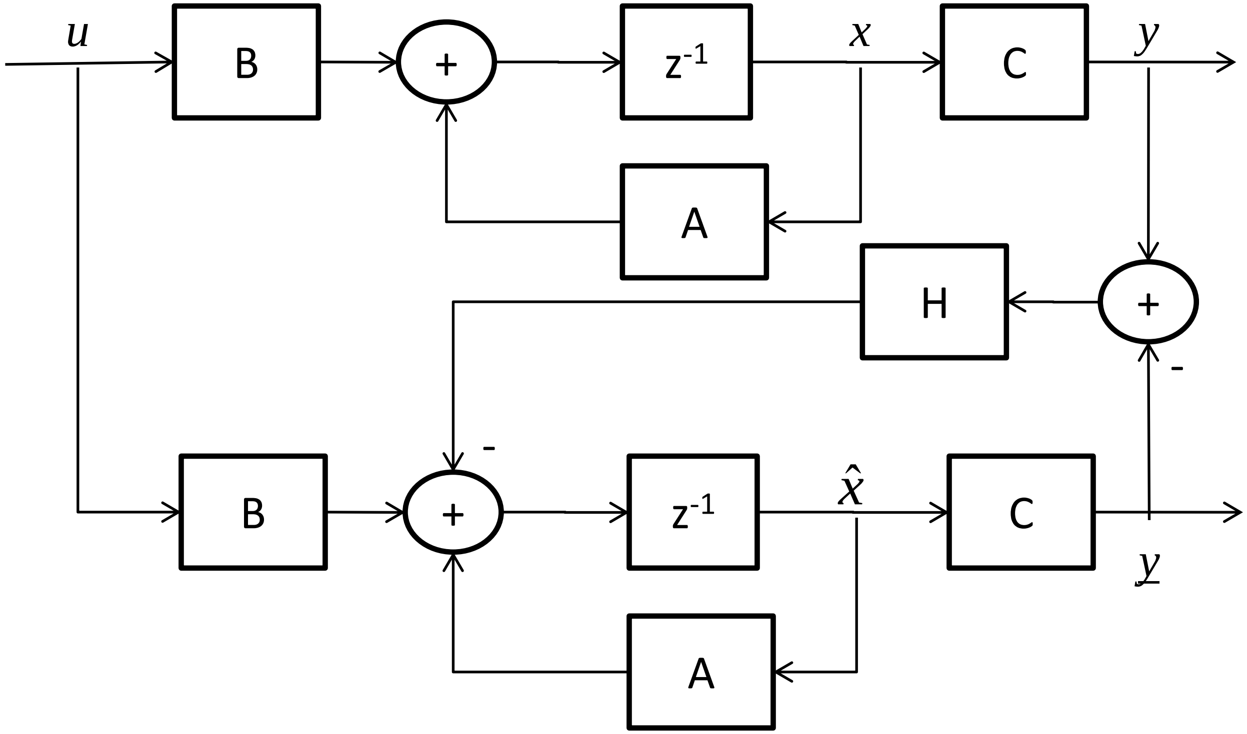
- Observer canonical system is observable.

verify:

$$A = \begin{bmatrix} 0 & 0 & \dots & 0 & -a_n \\ 1 & 0 & \dots & 0 & -a_{n-1} \\ 0 & 1 & 0 & \dots & -a_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -a_1 \end{bmatrix} \quad B = \begin{bmatrix} b_n \\ b_{n-1} \\ b_{n-2} \\ \vdots \\ b_1 \end{bmatrix} \quad C = [0, \cdots 0, 1]$$

- All observerable systems can be transformed into the observer canonical form

Observers (State Estimators)



Let $e = \hat{x} - x$ then $e(k+1) = Ae(k)$
What if A is not stable?

Error Dynamics

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) - H(\hat{y}(k) - y(k))$$
$$\hat{y}(k) = C\hat{x}(k)$$

since $\hat{y}(k) - y(k) = Ce(k)$

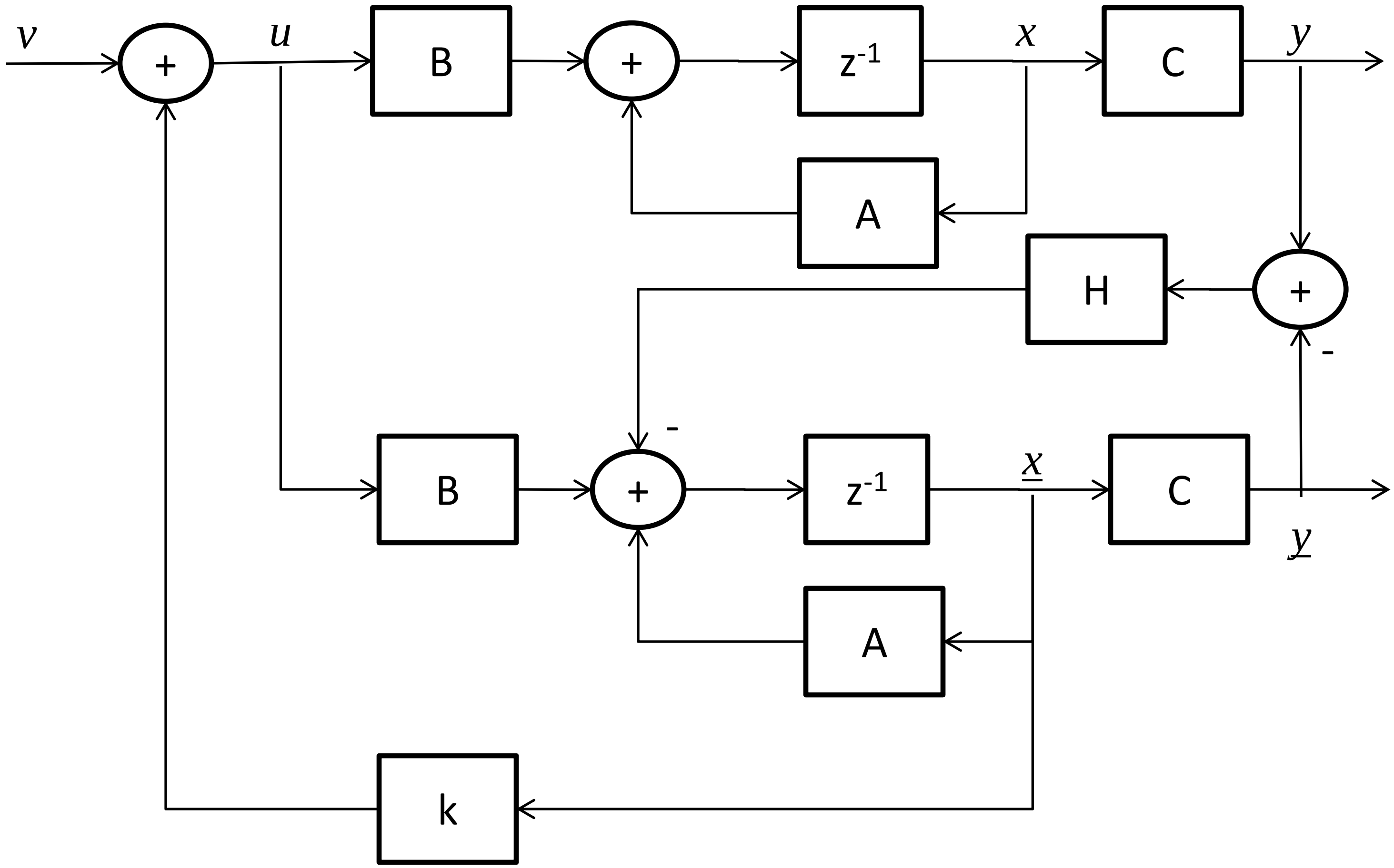
we get: $e(k+1) = [A - HC]e(k)$

note: C is a row vector, and H is column vector. Can we place the eigenvalues of [A-HC] arbitrarily?

What if {A, C} is observable? (Hint: what does HC look like?)

$$A = \begin{bmatrix} 0 & 0 & \dots & 0 & -a_n \\ 1 & 0 & \dots & 0 & -a_{n-1} \\ 0 & 1 & 0 & \dots & -a_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -a_1 \end{bmatrix} \quad B = \begin{bmatrix} b_n \\ b_{n-1} \\ b_{n-2} \\ \vdots \\ b_1 \end{bmatrix} \quad C = [0, \cdots 0, 1]$$

Complete Picture



Next lecture: controller design