Today's EE 233 Lecture

- Simulation diagrams.
- Flow graphs.
- Extracting state-space models.
- Similarity transformation and transfer functions.
- Summary.

Simulation Graphs

- Discrete-time systems may be represented by a difference equation or a transfer function.

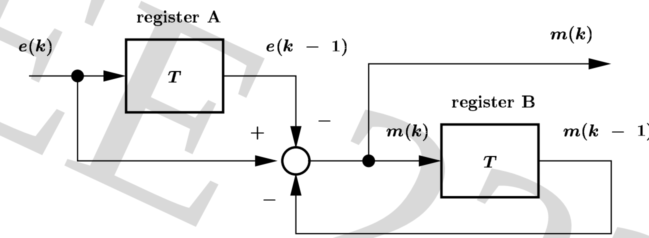
  It can also be represented by simulation diagrams.

- Basic element is a shift register.

The value appearing at the input is shifted into the register every T seconds. The number that is currently stored is shifted out.

Simulation Graphs

- Solves the difference equation.



Set register A to $e(0)$ and
set register B to the initial condition $m(0)$.

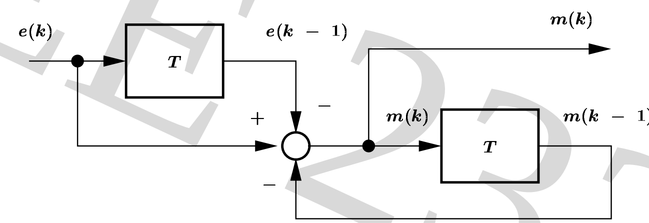
Output $m(k)$ will be the solution to

$$m(k) = e(k) - e(k-1) - m(k-1)$$

Simulation Graphs

- Example 1. Representing difference equations.

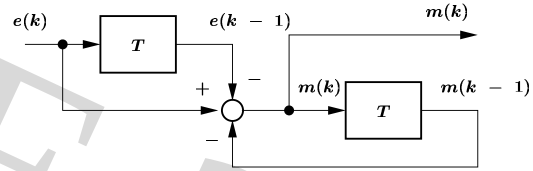
$$m(k) = e(k) - e(k-1) - m(k-1)$$

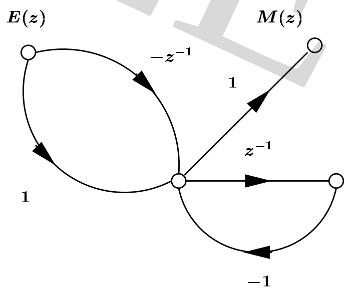


Interconnecting multiplication and summing devices with the delay element (shift register) allows representation of LTI difference equation.

Simulation Graphs

- Example 2. Convert to SFG.





Recall Mason's gain formula which gives

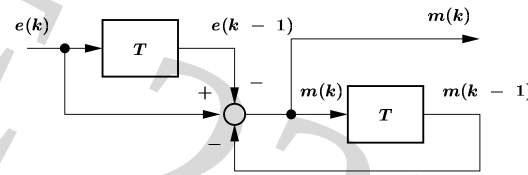
$$\frac{M(z)}{E(z)} = \frac{1 - z^{-1}}{1 + z^{-1}} = \frac{z - 1}{z + 1}$$

Simulation Graphs

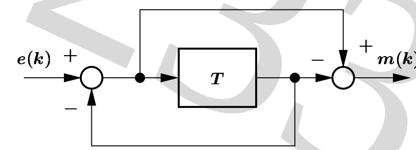
- Example 3. Take the first-order system

$$m(k) = e(k) - e(k-1) - m(k-1)$$

A nonminimal representation.



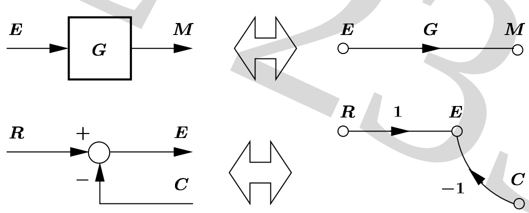
A minimal representation.



Simulation Graphs

- Analog simulation in continuous-time uses integrators. In discrete-time, the basic block is a time delay.

- Signal flow graphs can also be used for graphical representation of a discrete-time system.



Simulation Graphs

- Many ways of representing a system.
  - transfer function
  - block diagram and signal flow graph

- Block diagram and signal flow graph representations of a system is not unique.

  Recall, an nth-order continuous-time system can be represented with n integrators.

  A minimal representation of an nth-order discrete-time system contains n time delay elements.

- The system state at the $(k + 1)$th time instant is

$$x(k + 1) = f[x(k), u(k)]$$

where $x(k)$ is the current state and $u(k)$ is the current input.

- The system output response is

$$y(k) = g[x(k), u(k)]$$

- Example 4. Find the state variable representation for a system described by the difference equation

$$y(k + 2) = u(k) + 1.7y(k + 1) - 0.72y(k)$$

- Let

$$\begin{aligned} x_1(k) &= y(k) \\ x_2(k) &= x_1(k + 1) = y(k + 1) \end{aligned}$$

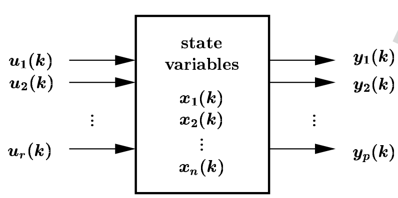
Then

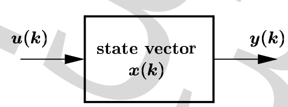
$$\begin{aligned} x_2(k + 1) &= y(k + 2) \\ &= u(k) + 1.7y(k + 1) - 0.72y(k) \end{aligned}$$

- Transfer function representation for LTI DT systems.

$$M(z) = G(z)E(z)$$

- State variable representation.





- For a linear system,

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

where $A(k)$, $B(k)$, $C(k)$ and $D(k)$ are time-varying n x n, n x r, p x n and p x r matrices, respectively.

- If the system is LTI,

$$\begin{aligned} x(k + 1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

- Do we have to go through explicitly figuring out how to assign the state variables every time we need a state-space form?

  Is there a standard procedure or form we can follow?

- We know we can go from the difference equation to the transfer function by z-transform.

  Consider the transfer function

$$G(z) = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \ldots + b_0}{z^n + a_{n-1}z^{n-1} + \ldots + a_0}$$

- Noting the real transform property, we can assign state variables as

$$\begin{aligned} E(z) &\rightarrow e(k) \stackrel{\triangle}{=} x_1(k) \\ zE(z) &\rightarrow e(k+1) = x_1(k+1) \stackrel{\triangle}{=} x_2(k) \\ z^2E(z) &\rightarrow e(k+2) = x_2(k+1) \stackrel{\triangle}{=} x_3(k) \\ &\vdots \\ z^{n-1}E(z) &\rightarrow e(k+n-1) = x_{n-1}(k+1) \stackrel{\triangle}{=} x_n(k) \\ z^nE(z) &\rightarrow e(k+n) = x_n(k+1) \end{aligned}$$

  We now have the state equations for $x_i(k+1)$, $i = 1, \ldots, n-1$ in terms of other state variables.

- Writing in matrix form,

$$x(k+1) = \begin{bmatrix} 0 & 1 \\ -0.72 & 1.7 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$
$$y(k) = [1\ 0]x(k)$$

- State-space representation is crucial to the analysis of control systems.

  Modern control system analysis techniques usually start by expressing system models in state-space form.

  Easy to change from state-space form to other forms (such as transfer function or difference equations).

- Introducing a dummy variable $E(z)$ and using $G(z) = Y(z)/U(z)$,

$$\frac{Y(z)}{U(z)} = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \ldots + b_0}{z^n + a_{n-1}z^{n-1} + \ldots + a_0} \cdot \frac{E(z)}{E(z)}$$

- Splitting the above equation gives

$$\begin{aligned} Y(z) &= (b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \ldots + b_0)E(z) \\ U(z) &= (z^n + a_{n-1}z^{n-1} + \ldots + a_0)E(z) \end{aligned}$$

State Variables

- Consider again the transfer function

$$G(z) = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \ldots + b_0}{z^n + a_{n-1}z^{n-1} + \ldots + a_0}$$

Multiplying by z^{-n}/z^{-n}, we can write

$$\frac{Y(z)}{U(z)} = \frac{b_{n-1}z^{-1} + b_{n-2}z^{-2} + \ldots + b_0z^{-n}}{1 + a_{n-1}z^{-1} + \ldots + a_0z^{-n}} \cdot \frac{E(z)}{E(z)}$$

We can split this into two equations,

$$Y(z) = (b_{n-1}z^{-1} + b_{n-2}z^{-2} + \ldots + b_0z^{-n})E(z)$$
$$U(z) = (1 + a_{n-1}z^{-1} + \ldots + a_0z^{-n})E(z)$$

$$\Rightarrow E(z) = U(z) - a_{n-1}z^{-1}E(z) - \ldots - a_0z^{-n}E(z)$$

State Variables

- Expanding and taking the inverse z-transform of

$$U(z) = (z^n + a_{n-1}z^{n-1} + \ldots + a_0)E(z)$$

gives us the state equation for $x_n(k+1)$.

$$x_n(k+1) = -a_0x_1(k) - a_1x_2(k) - \ldots - a_{n-1}x_n(k) + u(k)$$

The rest of the state equations are simply,

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_3(k) \\ &\vdots \\ x_{n-1}(k+1) &= x_n(k) \end{aligned}$$

State Variables

- In matrix form,

$$x(k+1) = Ax(k) + Bu(k)$$

where

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} \Rightarrow x(k+1) = \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ & & \vdots & & \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

State Variables

- The output equation is obtained by expanding and taking the inverse z-transform of

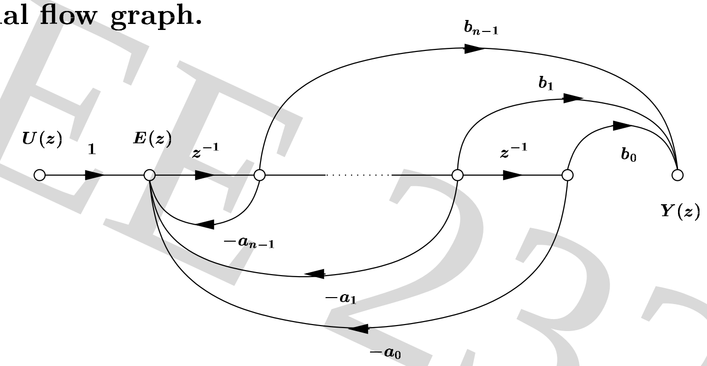
$$Y(z) = (b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \ldots + b_0)E(z)$$

which gives in matrix form,

$$y(k) = [b_0\ b_1\ \ldots\ b_{n-1}] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}$$

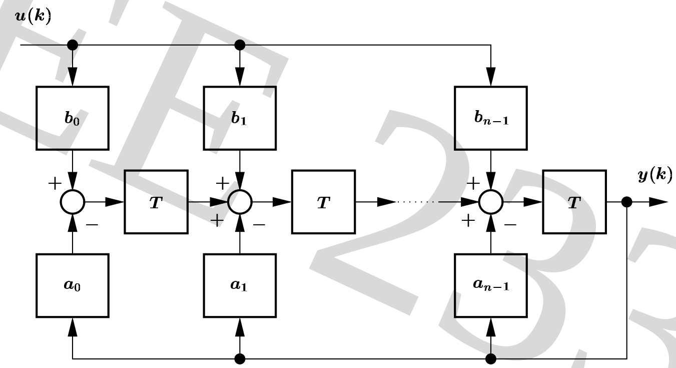
or $y(k) = Cx(k)$.

- Signal flow graph.

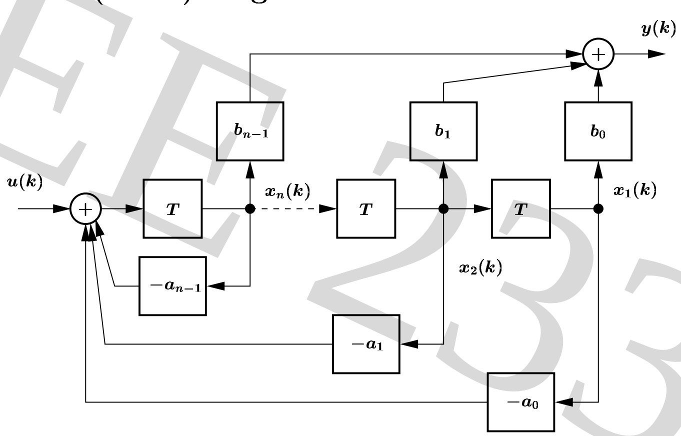


$$E(z) = U(z) - a_{n-1}z^{-1}E(z) - \ldots - a_0 z^{-n}E(z)$$
$$Y(z) = (b_{n-1}z^{-1} + b_{n-2}z^{-2} + \ldots + b_0 z^{-n})E(z)$$

- Observable canonical form block diagram.



- Simulation (block) diagram.



- The resulting state-space form previously considered is commonly called the controllable canonical form.

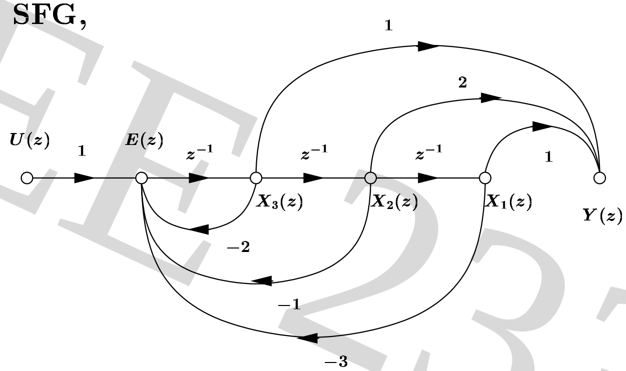
  Another standard form is the observable canonical form.

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k)$$

$$A = \begin{bmatrix} 0\ 0\ 0 \ldots 0 & -a_0 \\ 1\ 0\ 0 \ldots 0 & -a_1 \\ 0\ 1\ 0 \ldots 0 & -a_2 \\ \vdots & \\ 0\ 0\ 0 \ldots 1 & -a_{n-1} \end{bmatrix} \qquad B = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix}$$

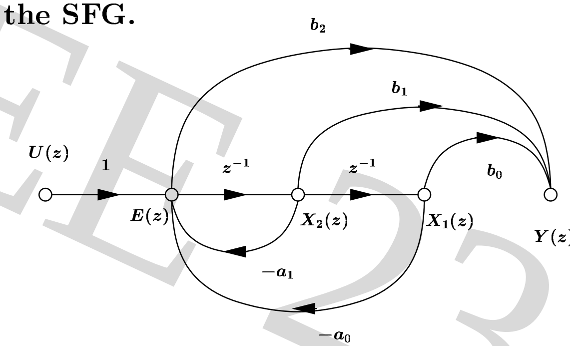
$$C = [0\ 0\ \ldots\ 0\ 1]$$

- For the SFG,



$$U(z) = (1 + 2z^{-1} + z^{-2} + 3z^{-3})E(z)$$
$$Y(z) = (z^{-1} + 2z^{-2} + z^{-3})E(z)$$

- Look at the SFG.



State equations.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

- Example 5. Derive the SFG and state equations for

$$G(z) = \frac{Y(z)}{U(z)} = \frac{z^2 + 2z + 1}{z^3 + 2z^2 + z + 3}$$

Standard form using controllable canonical form.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & -1 & -2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1\ 2\ 1] \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}$$

- Example 6. Derive the SFG and state equations for

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_2 z^2 + b_1 z + b_0}{z^2 + a_1 z + a_0}$$

Note that the numerator and the denominator have the same order. Using a similar technique as before,

$$\frac{Y(z)}{U(z)} = \frac{b_2 + b_1 z^{-1} + b_0 z^{-2}}{1 + a_1 z^{-1} + a_0 z^{-2}} \cdot \frac{E(z)}{E(z)}$$

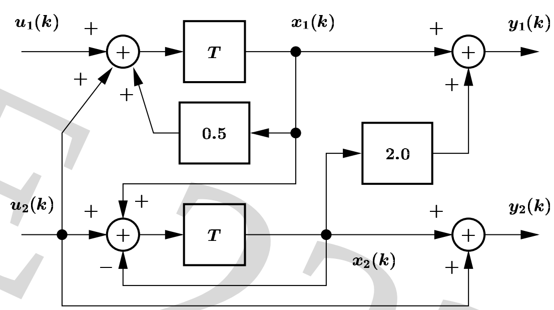
which gives

$$E(z) = U(z) - a_1 z^{-1} E(z) - a_0 z^{-2} E(z)$$
$$Y(z) = \underbrace{b_2 E(z)} + b_1 z^{-1} E(z) + b_0 z^{-2} E(z)$$

how do you handle this?

State Variables

- Example 7. Determine the state equations from the block diagram.



$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

Other State Variable Formulations

- Different representations for SISO DT systems.
  - difference equation.
  - transfer function.
  - state-space form.
  - block diagram and SFG.

- Specific applications.
  - transfer function is important for control design based on assigning poles and zeros.
  - state-space form is good for simulations and looking at internal variables.
  - block diagram and SFG are useful for visualizing signal flow within the system.

State Variables

- For the output equation, use the SFG to get

$$Y(z) = b_0X_1(z) + b_1X_2(z) + b_2E(z)$$
$$E(z) = U(z) - a_0X_1(z) - a_1X_2(z)$$

Eliminating $E(z)$ gives

$$Y(z) = b_2U(z) + (b_0 - b_2a_0)X_1(z) + (b_1 - b_2a_1)X_2(z)$$

Finally, the output equation is

$$y(k) = [(b_0 - b_2a_0)\ (b_1 - b_2a_1)] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + b_2u(k)$$

A transfer function with numerator and denominator of the same orders may be handled in a similar manner.

State Variables

- State models may be derived from the a SFG or a block diagrams by
  - draw the SFG or block diagram.
  - assign a state variable to each delay output.
  - write the equation for each delay input and each system output using only the delay outputs and the system input.

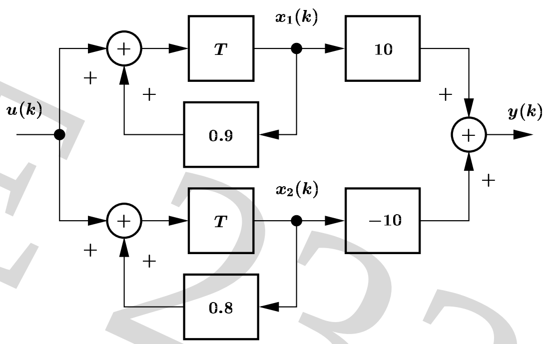
- The transfer function is an input-output system model.

  The state-space model gives an input-output relationship along with an internal description of a system.

  Which one is unique and which one is not?

Other State Variable Formulations

- Simulation diagram (parallel form).



$$x(k+1) = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.8 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$
$$y(k) = [10 \ \ -10]x(k)$$

Other State Variable Formulations

- Example 8. Consider

$$y(k+2) = u(k) - 1.7y(k+1) - 0.72y(k)$$

Taking the z-transform to get the transfer function

$$\frac{Y(z)}{U(z)} = \frac{1}{z^2 + 1.7z + 0.72}$$

The transfer function can be expressed either as

$$\frac{Y(z)}{U(z)} = \frac{10}{z - 0.9} + \frac{-10}{z - 0.8} \quad \text{parallel form}$$

or

$$\frac{Y(z)}{U(z)} = \left[\frac{1}{z - 0.9}\right]\left[\frac{1}{z - 0.8}\right] \quad \text{series form}$$

Similarity Transformations

- Different state models exist for a given system?

  How can we derive new (different) state models given a state model for a system.

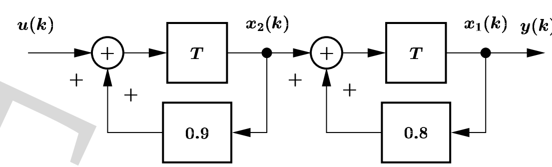
- Consider the state equation

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

- Apply a linear transformation on the state vector, i.e., $x(k) \rightarrow w(k)$.

Other State Variable Formulations

- Simulation diagram (series form).



$$x(k+1) = \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$
$$y(k) = [1 \ 0]x(k)$$

- General forms.

  parallel : $G(z) = G_{1p}(z) + G_{2p}(z) + \ldots + G_{np}(z)$

  series : $G(z) = G_{1s}(z) \cdot G_{2s}(z) \cdot \ldots \cdot G_{ns}(z)$

Similarity Transformations

- The original state equations may now be written as

$$w(k+1) = P^{-1}APw(k) + P^{-1}Bu(k)$$
$$y(k) = CPw(k) + Du(k)$$

or

$$w(k+1) = A_w w(k) + B_w u(k)$$
$$y(k) = C_w w(k) + D_w u(k)$$

where

$$A_w = P^{-1}AP \qquad B_w = P^{-1}B$$
$$C_w = CP \qquad D_w = D$$

For each non-singular P, we have a different state model for the same system.

Similarity Transformations

- Properties of the similarity transformation.
  - The determinant, trace and eigenvalues of the matrix are invariant under the transformation.
  - The transfer function is also unchanged.

$$C[zI - A]^{-1}B + D = C_w[zI - A_w]^{-1}B_w + D_w$$

- Example 9. Consider

$$x(k+1) = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.8 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$
$$y(k) = [10 \ \ -10]x(k)$$

Similarity Transformations

- Express the original states as linear combinations of the new states.

$$\begin{aligned} x_1(k) &= p_{11}w_1(k) + p_{12}w_2(k) + \ldots + p_{1n}w_n(k) \\ x_2(k) &= p_{21}w_1(k) + p_{22}w_2(k) + \ldots + p_{2n}w_n(k) \\ &\vdots \\ x_n(k) &= p_{n1}w_1(k) + p_{n2}w_2(k) + \ldots + p_{nn}w_n(k) \end{aligned}$$

or concisely,

$$x(k) = Pw(k)$$

where P is an nxn non-singular matrix and $w(k)$ is the new state vector.

Similarity Transformations

- The characteristic equation of A is

$$|zI - A| = 0$$

The eigenvalues z_i are the roots of the characteristic equation, i.e.,

$$|zI - A| = (s - z_1)(s - z_2)\ldots(s - z_n) = 0$$

- Under the similarity transformation $A_w = P^{-1}AP$,

$$\begin{aligned} |zI - A_w| &= |zI - P^{-1}AP| = |zP^{-1}IP - P^{-1}AP| \\ &= |P^{-1}||zI - A||P| \\ &= |zI - A| \end{aligned}$$

Similarity Transformations

- If the eigenvalues of the system are distinct, we can come up with a similarity transformation to diagonalize the system.

- If A is nxn, consider an n-vector m_i and a scalar z_i such that

$$Am_i = z_i m_i$$

We can write

$$(z_i I - A)m_i = 0$$

For a nontrivial solution, $|z_i I - A| = 0$.

Similarity Transformations

- Example 10. Consider

$$x(k+1) = \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$
$$y(k) = [1\ 0]x(k)$$

- We can use the Matlab `eig` function to determine the eigenvalues and eigenvectors.

```
>> [M,Lambda] = eig([0.8 1; 0 0.9])
```

$$M = \begin{bmatrix} 1 & 1 \\ 0 & 0.1 \end{bmatrix} \Rightarrow M^{-1} = \begin{bmatrix} 1 & -10 \\ 0 & 10 \end{bmatrix}$$

Similarity Transformations

- Select a similarity transformation

$$P = \begin{bmatrix} 0 & 1 \\ 1 & -10 \end{bmatrix} \Rightarrow P^{-1} = \begin{bmatrix} 10 & 1 \\ 1 & 0 \end{bmatrix}$$

- This gives the new state-space model

$$A_w = \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix},\ B_w = \begin{bmatrix} 11 \\ 1 \end{bmatrix},\ C_w = [-10\ 110]$$

$$w(k+1) = \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} w(k) + \begin{bmatrix} 11 \\ 1 \end{bmatrix} u(k)$$
$$y(k) = [-10\ 110]w(k)$$

Similarity Transformations

- If z_i are distinct eigenvalues and with corresponding eigenvectors m_i, we can form the matrix equation

$$A[m_1\ m_2\ \ldots\ m_n] = [m_1\ m_2\ \ldots\ m_n] \begin{bmatrix} z_1 & 0 & \ldots & 0 \\ 0 & z_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & z_n \end{bmatrix}$$

or $AM = M\Lambda$.

- M is called the modal matrix. The inverse of M exists if we have linearly independent eigenvectors z_i. Thus, we have a similarity transformation $P = M$ that gives

$$\Lambda = M^{-1}AM$$

Transfer Functions

- Extracted state-space model from the transfer function.
  - using SFG or block diagrams.
  - using z-transforms.
  - standard forms from difference equation.

- How do we get the transfer function from the state-space model?
  - use SFG or block diagrams with Mason's gain rule.
  - using z-transforms.

Transfer Functions

- Using z-transform directly on the state equations.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ \Rightarrow zX(z) - zx(0) &= AX(z) + BU(z) \end{aligned}$$

- Ignoring initial conditions and collecting terms.

$$[zI - A]X(z) = BU(z)$$

Solving for $X(z)$ results in

$$X(z) + [zI - A]^{-1}BU(z)$$

Similarity Transformations

- Using the $P = M$ for the similarity transformation,

$$A_w = \Lambda = M^{-1}AM = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.9 \end{bmatrix}$$

$$B_w = M^{-1}B = \begin{bmatrix} -10 \\ 1 \end{bmatrix}$$

$$C_w = CM = [1\ 10]$$

- The new state model is

$$w(k+1) = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.9 \end{bmatrix} w(k) + \begin{bmatrix} -10 \\ 1 \end{bmatrix} u(k)$$

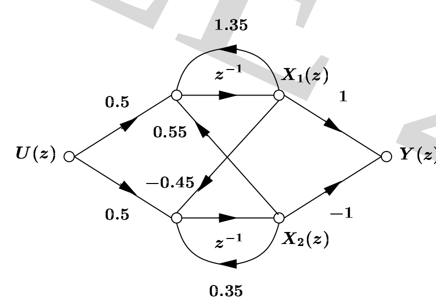
$$y(k) = [1\ 10]w(k)$$

Transfer Functions

- Example 11. Derive the transfer function.

$$x(k+1) = \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} x(k) + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} u(k)$$

$$y(k) = [1\ -1]x(k)$$



Mason's gain rule gives

$$\frac{Y(z)}{U(z)} = \frac{1}{z^2 - 1.7z + 0.72}$$

Transfer Functions

- Example 12. Consider the previous example.

$$x(k+1) = \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} x(k) + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} u(k)$$
$$y(k) = [1 \ -1]x(k)$$

- First compute $[zI - A]^{-1}$.

$$[zI - A] = \begin{bmatrix} z - 1.35 & -0.55 \\ 0.45 & z - 0.35 \end{bmatrix}$$

$$[zI - A]^{-1} = \frac{1}{z^2 - 1.7z + 0.72} \begin{bmatrix} z - 0.35 & 0.55 \\ -0.45 & z - 1.35 \end{bmatrix}$$

Summary

- Simulation diagrams and signal flow graphs.
- Extracting state-space models.
- Standard forms from difference equations.
- Similarity transformation and transfer functions.

Transfer Functions

- From the output equation.

$$y(k) = Cx(k) + Du(k)$$
$$\Rightarrow Y(z) = CX(z) + DU(z)$$

- Substituting the $X(z)$ equation gives

$$Y(z) = \left[C[zI - A]^{-1}B + D\right] U(z)$$

The system transfer function is

$$G(z) = C[zI - A]^{-1}B + D$$

Transfer Functions

- Forming the transfer function.

$$G(z) = C[zI - A]^{-1}B$$
$$= \frac{[1 \ 1]}{z^2 - 1.7z + 0.72} \begin{bmatrix} z - 0.35 & 0.55 \\ -0.45 & z - 1.35 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$
$$= \frac{[1 \ 1]}{z^2 - 1.7z + 0.72} \begin{bmatrix} 0.5z + 0.1 \\ 0.5z - 0.9 \end{bmatrix}$$
$$= \frac{1}{z^2 - 1.7z + 0.72}$$