Chapter 4
DISCRETE-TIME SYSTEMS
4.8 State-Space Representation

Victoria, BC, Canada
Email: aantoniou@ieee.org

July 14, 2018

State-Space Representation

★ Given a discrete-time network or signal flow graph, a corresponding state-space representation can be deduced.

State-Space Representation

- Given a discrete-time network or signal flow graph, a corresponding state-space representation can be deduced.
- A state-space representation provides another way of finding the time-domain response of a discrete-time system.

Derivation

Let us consider an arbitrary discrete-time system with the following properties:

– It contains N unit delays.

– Each and every loop in the network includes at least one unit delay.

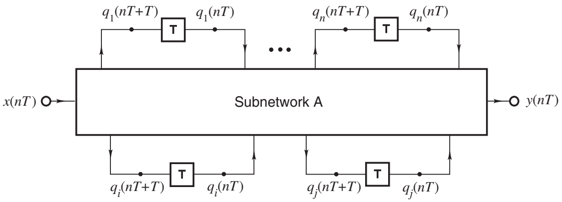
Derivation

Let us consider an arbitrary discrete-time system with the following properties:

- It contains N unit delays.
- Each and every loop in the network includes at least one unit delay.

The second condition will ensure that the signal flow graph of the system is *computable* and, therefore, realizable in terms of unit delays, adders, and multipliers.

See Sec. 2.8.1 for details about computability.



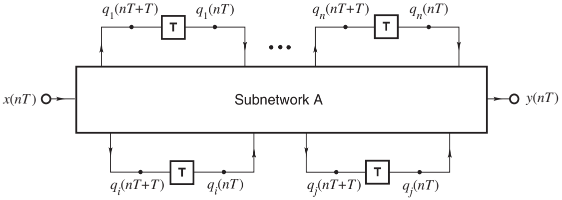
(*a*)

★ Let us assign variables

$$q_i(nT) \qquad \text{for} \quad i = 1, 2, \ldots, N$$

at the outputs of the N unit delays.

Derivation *Cont'd*



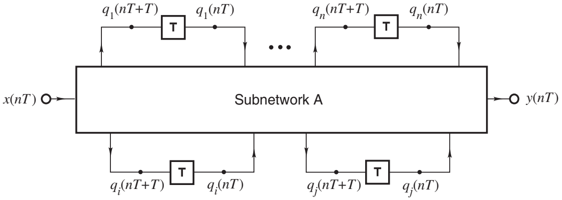
(*a*)

- Let us assign variables

$$q_i(nT) \qquad \text{for} \quad i = 1, 2, \ldots, N$$

at the outputs of the N unit delays.

- These variables represent stored quantities and can be referred to as *state variables*.



(*a*)

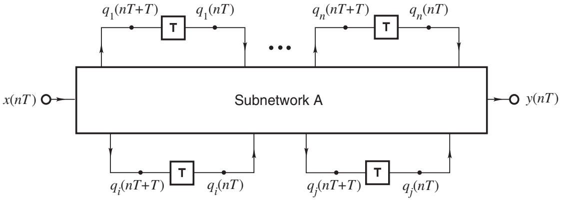
- Let us assign variables

$$q_i(nT) \qquad \text{for} \quad i = 1, 2, \ldots, N$$

at the outputs of the N unit delays.

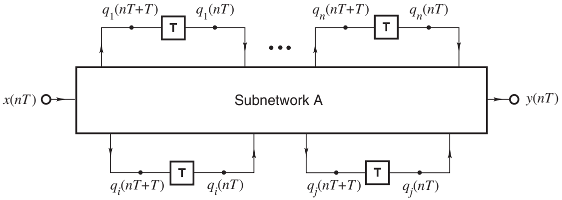
- These variables represent stored quantities and can be referred to as *state variables*.

- The signals at the inputs of the unit delays can, obviously, be represented by corresponding variables $q_i(nT + T)$.



(*a*)

★ Let us apply a signal $x(nT)$ at the input of the system assuming that all the state variables are zero.



(*a*)

★ Let us apply a signal $x(nT)$ at the input of the system assuming that all the state variables are zero.

★ The response produced at the input of the ith unit delay can be determined by applying Mason's gain formula, i.e.,

$$q_i(nT+T) = \frac{1}{\Delta}\left(\sum_k T_k \Delta_k\right) x(nT)$$

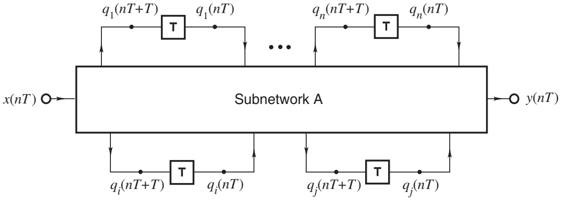
to the signal flow graph of subnetwork A.

...

$$q_i(nT + T) = \frac{1}{\Delta}\left(\sum_k T_k\Delta_k\right)x(nT)$$

Recall that

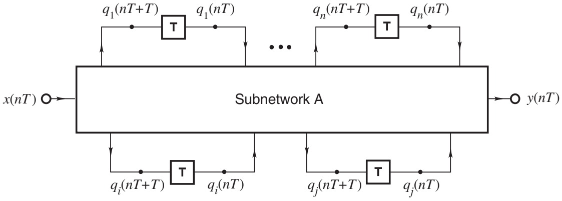
- T_k is the transmittance of the kth direct path between the system input and the input of the ith unit delay,
- Δ is the determinant of the signal flow graph, and
- Δ_k is the determinant of the subgraph that does not touch the kth direct path between the system input and the input of the ith unit delay.



(a)

We note the following:

★ Since each and every loop includes at least one unit delay, the total number of loops cannot be larger than N.



(a)

We note the following:

- ★ Since each and every loop includes at least one unit delay, the total number of loops cannot be larger than N.
- ★ Hence all the loops will be broken if the unit delays are removed, i.e., *there are no loops inside subnetwork A*.

Derivation *Cont'd*

The determinants in Mason's gain formula are given by

$$\begin{aligned}\Delta &= 1 - \sum_u L_{u1} + \sum_v P_{v2} - \sum_w P_{w3} + \cdots \\ \Delta_k &= 1 - \sum_u L'_{u1} + \sum_v P'_{v2} - \sum_w P'_{w3} + \cdots\end{aligned}$$

where L_{u1} and $L'_{u1}$are loop transmittances, P_{v2} and P'_{v2} are products of pairs of loop transmittances, P_{w3} and $P'_{w3}$are products of triplets of loop transmittances, etc.

The determinants in Mason's gain formula are given by

$$\begin{aligned}\Delta &= 1 - \sum_u L_{u1} + \sum_v P_{v2} - \sum_w P_{w3} + \cdots \\ \Delta_k &= 1 - \sum_u L'_{u1} + \sum_v P'_{v2} - \sum_w P'_{w3} + \cdots\end{aligned}$$

where L_{u1} and $L'_{u1}$are loop transmittances, P_{v2} and P'_{v2} are products of pairs of loop transmittances, P_{w3} and $P'_{w3}$are products of triplets of loop transmittances, etc.

- ★ Since there are no loops inside subnetwork A, we have

$$\Delta = \Delta_k = 1$$

The determinants in Mason's gain formula are given by

$$\Delta = 1 - \sum_u L_{u1} + \sum_v P_{v2} - \sum_w P_{w3} + \cdots$$
$$\Delta_k = 1 - \sum_u L'_{u1} + \sum_v P'_{v2} - \sum_w P'_{w3} + \cdots$$

where L_{u1} and $L'_{u1}$are loop transmittances, P_{v2} and P'_{v2} are products of pairs of loop transmittances, P_{w3} and $P'_{w3}$are products of triplets of loop transmittances, etc.

- Since there are no loops inside subnetwork A, we have

$$\Delta = \Delta_k = 1$$

- Also all transmittances T_k are independent of the shift operator $\mathcal{E}^{-1}$, i.e., they must be constants that depend on the multiplier constants inside subnetwork A.

★ Therefore, the response produced at the input of the ith unit delay by a nonzero input $x(nT)$, i.e.,

$$q_i(nT + T) = \frac{1}{\Delta}\left(\sum_k T_k \Delta_k\right) x(nT)$$

can be expressed as

$$q_i(nT + T) = b_i x(nT) \qquad \text{for} \quad i = 1, 2, \ldots, N \qquad \text{(A)}$$

where $b_1, b_2, \ldots, b_N$ are constants which are independent of nT for a time-invariant discrete-time system.

★ Similarly, if input $x(nT)$ and all the state variables except the jth one are zero, we have

$$q_i(nT + T) = a_{ij} q_j(nT) \quad \text{for} \quad i = 1, 2, \ldots, N \qquad \text{(B)}$$

where $a_1, a_2, \ldots, a_N$ are constants which are independent of nT for a time-invariant digital system.

★ Now if the system is linear, the response at the input of the ith unit delay is obtained from Eqs. (A) and (B) as

$$q_i(nT+T) = \sum_{j=1}^{N} a_{ij} q_j(nT) + b_i x(nT) \quad \text{for} \quad i = 1, 2, \ldots, N \tag{C}$$

by applying the principle of superposition.

★ Now if the system is linear, the response at the input of the ith unit delay is obtained from Eqs. (A) and (B) as

$$q_i(nT + T) = \sum_{j=1}^{N} a_{ij} q_j(nT) + b_i x(nT) \quad \text{for} \quad i = 1,\ 2,\ \ldots,\ N \tag{C}$$

by applying the principle of superposition.

★ Similarly, the response at the output of the system, $y(nT)$, due to input excitation $x(nT)$ and state variables $q_j(nT)$ for $j = 1,\ 2,\ \ldots,\ N$ can be expressed as

$$y(nT) = \sum_{j=1}^{N} c_j q_j(nT) + d_0 x(nT) \tag{D}$$

★ Summarizing the results obtained so far, the N-delay network we started with can be represented by the state-space equations

$$q_i(nT+T) = \sum_{j=1}^{N} a_{ij} q_j(nT) + b_i x(nT) \quad \text{for} \quad i = 1, 2, \ldots, N \quad \text{(C)}$$

$$y(nT) = \sum_{j=1}^{N} c_j q_j(nT) + d_0 x(nT) \quad \text{(D)}$$

★ Summarizing the results obtained so far, the N-delay network we started with can be represented by the state-space equations

$$q_i(nT+T) = \sum_{j=1}^{N} a_{ij} q_j(nT) + b_i x(nT) \quad \text{for} \quad i = 1, 2, \ldots, N \quad \text{(C)}$$

$$y(nT) = \sum_{j=1}^{N} c_j q_j(nT) + d_0 x(nT) \quad \text{(D)}$$

★ These equations can now be expressed in matrix form as

$$\begin{aligned} \mathbf{q}(nT+T) &= \mathbf{A}q(nT) + \mathbf{b}x(nT) \\ y(nT) &= \mathbf{c}^T\mathbf{q}(nT) + dx(nT) \end{aligned}$$

Derivation *Cont'd*

...

$$\mathbf{q}(nT+T) = \mathbf{A}q(nT) + \mathbf{b}x(nT)$$
$$y(nT) = \mathbf{c}^T\mathbf{q}(nT) + dx(nT)$$

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}$$

$$\mathbf{c}^T = \begin{bmatrix} c_1 & c_2 & \cdots & c_N \end{bmatrix}, \quad d = d_0$$

and

$$\mathbf{q}(nT) = \begin{bmatrix} q_1(nT) & q_2(nT) & \cdots & q_N(nT) \end{bmatrix}^T$$

is a column vector whose elements are the state variables of the discrete-time system network.

- Note that the choice of state variables is not unique.

  For example, one can assign state variables $q_1(nT)$, $q_2(nT)$, $\ldots$, $q_N(nT)$ to nodes in any order, and each choice will give a valid state-space representation.

- Note that the choice of state variables is not unique.

  For example, one can assign state variables $q_1(nT), q_2(nT), \ldots, q_N(nT)$ to nodes in any order, and each choice will give a valid state-space representation.

- In fact, given a set of state variables

$$q_1(nT), q_2(nT), \ldots, q_N(nT)$$

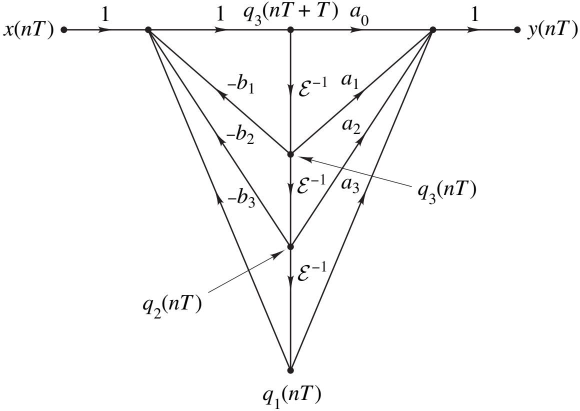
  which can be represented by column vector $\mathbf{q}$, another valid set of state variables can be readily obtained by applying a transformation of the form

$$\tilde{\mathbf{q}}(nT) = \mathbf{M}\mathbf{q}(nT)$$

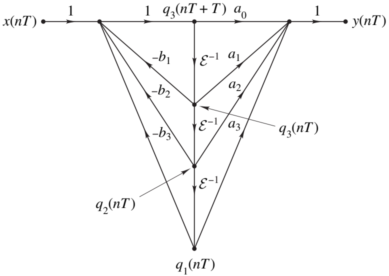
  where $\mathbf{M}$ is an $N \times N$ matrix.

Example

Obtain a state-space representation for the system represented by the signal flow chart shown.



Solution One possible assignment of state variables is shown in the figure.

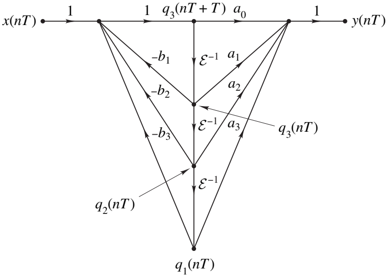


We have

$$\begin{aligned} q_1(nT+T) &= q_2(nT) \\ q_2(nT+T) &= q_3(nT) \\ q_3(nT+T) &= -b_3 q_1(nT) - b_2 q_2(nT) - b_1 q_3(nT) + x(nT) \end{aligned}$$

Example *Cont'd*

...



$$q_1(nT+T) = q_2(nT)$$
$$q_2(nT+T) = q_3(nT)$$
$$q_3(nT+T) = -b_3q_1(nT) - b_2q_2(nT) - b_1q_3(nT) + x(nT)$$

The output of the system can be expressed as

$$y(nT) = a_3q_1(nT) + a_2q_2(nT) + a_1q_3(nT) + a_0q_3(nT+T)$$

...

$$
\begin{aligned}
q_1(nT+T) &= q_2(nT) \\
q_2(nT+T) &= q_3(nT) \\
q_3(nT+T) &= -b_3 q_1(nT) - b_2 q_2(nT) - b_1 q_3(nT) + x(nT)
\end{aligned}
$$

The output of the system can be expressed as

$$y(nT) = a_3 q_1(nT) + a_2 q_2(nT) + a_1 q_3(nT) + a_0 q_3(nT+T)$$

Now if we eliminate $q_3(nT+T)$ in $y(nT)$, we get

$$
\begin{aligned}
y(nT) = {} & (a_3 - a_0 b_3) q_1(nT) + (a_2 - a_0 b_2) q_2(nT) \\
& + (a_1 - a_0 b_1) q_3(nT) + a_0 x(nT)
\end{aligned}
$$

Example *Cont'd*

The results obtained can now be expressed is matrix form as follows:

$$\begin{aligned}\mathbf{q}(nT + T) &= \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT)\\ y(nT) &= \mathbf{c}^T\mathbf{q}(nT) + dx(nT)\end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -b_3 & -b_2 & -b_1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{c}^T = \left[\, (a_3 - a_0 b_3)\ (a_2 - a_0 b_2)\ (a_1 - a_0 b_1) \,\right], \quad d = a_0$$

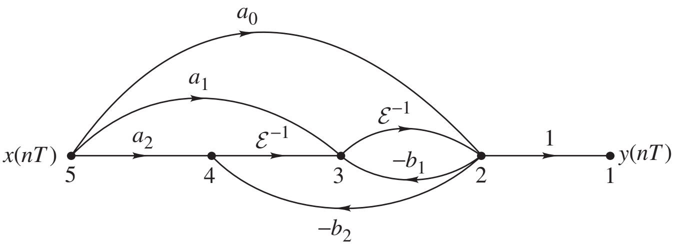
and

$$\mathbf{q}(nT) = \left[\, q_1(nT)\ q_2(nT)\ q_3(nT) \,\right]^T$$

Pitfall

Note: The state variables must *always* be defined at the outputs of the unit delays!

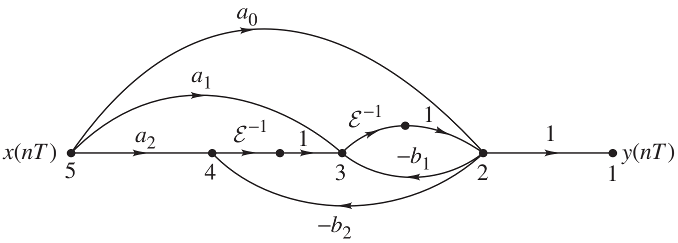
Nodes 2 and 3 in the signal flow graph of the figure below represent the outputs of the adders, *not* the outputs of the unit delays.



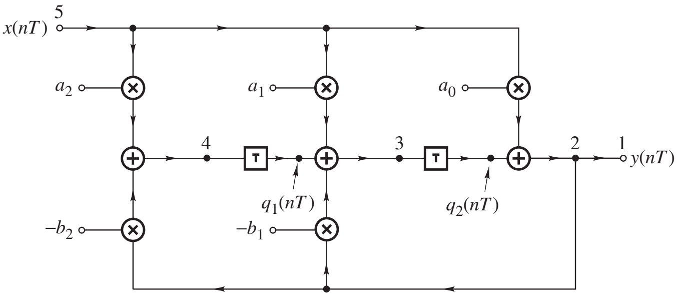
(*b*)

Pitfall *Cont'd*

The pitfall can be avoided by adding new nodes at the outputs of the unit delays as shown below.



Another way to avoid the problem is to use the network of the system instead of the signal flow graph.



Time-Domain Analysis

- The state-space characterization leads to a relatively simple alternative time-domain analysis.

Time-Domain Analysis

- The state-space characterization leads to a relatively simple alternative time-domain analysis.

- Evaluating $\mathbf{q}(nT+T)$ for $n = 0,\ 1,\ 2, \ldots,$ using the first state-space equation, i.e.,

$$\mathbf{q}(nT+T) = \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT)$$

we obtain

$$\begin{aligned}
\mathbf{q}(T) &= \mathbf{A}\mathbf{q}(0) + \mathbf{b}x(0) \\
\mathbf{q}(2T) &= \mathbf{A}\mathbf{q}(T) + \mathbf{b}x(T) \\
\mathbf{q}(3T) &= \mathbf{A}\mathbf{q}(2T) + \mathbf{b}x(2T) \\
&\vdots
\end{aligned}$$

...

$$\begin{aligned} \mathbf{q}(T) &= \mathbf{A}\mathbf{q}(0) + \mathbf{b}x(0) \\ \mathbf{q}(2T) &= \mathbf{A}\mathbf{q}(T) + \mathbf{b}x(T) \\ \mathbf{q}(3T) &= \mathbf{A}\mathbf{q}(2T) + \mathbf{b}x(2T) \\ &\vdots \end{aligned}$$

Hence

$$\begin{aligned} \mathbf{q}(2T) &= \mathbf{A}^2\mathbf{q}(0) + \mathbf{A}\mathbf{b}x(0) + \mathbf{b}x(T) \\ \mathbf{q}(3T) &= \mathbf{A}^3\mathbf{q}(0) + \mathbf{A}^2\mathbf{b}x(0) + \mathbf{A}\mathbf{b}x(T) + \mathbf{b}x(2T) \end{aligned}$$

In general,

$$\mathbf{q}(nT) = \mathbf{A}^n\mathbf{q}(0) + \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)}\mathbf{b}x(kT)$$

where $\mathbf{A}^0$ is the $N \times N$ identity matrix.

...

$$\mathbf{q}(nT) = \mathbf{A}^n\mathbf{q}(0) + \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)}\mathbf{b}x(kT)$$

If we now use the second state-space equation, i.e.,

$$y(nT) = \mathbf{c}^T\mathbf{q}(nT) + dx(nT)$$

we obtain the response of the system as

$$y(nT) = \mathbf{c}^T\mathbf{A}^n\mathbf{q}(0) + \mathbf{c}^T\sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)}\mathbf{b}x(kT) + dx(nT)$$

...

$$y(nT) = \mathbf{c}^T \mathbf{A}^n \mathbf{q}(0) + \mathbf{c}^T \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)} \mathbf{b} x(kT) + d x(nT)$$

If the system is initially relaxed then the state variables are all zero at time zero, i.e.,

$$\mathbf{q}(0) = 0$$

Thus for an initially relaxed system, we have

$$y(nT) = \mathbf{c}^T \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)} \mathbf{b} x(kT) + d x(nT)$$

Impulse Response

★ By letting $x(nT) = \delta(nT)$ in the general formula for the time-domain response, the impulse response $h(nT)$ of the system can be expressed as

$$h(nT) = \mathcal{R}\delta(t) = \mathbf{c}^T \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)} \mathbf{b}\delta(kT) + d\delta(nT)$$

which simplifies to

$$h(nT) = \begin{cases} d_0 & \text{for} \quad n = 0 \\ \mathbf{c}^T \mathbf{A}^{(n-1)} \mathbf{b} & \text{for} \quad n > 0 \end{cases}$$

Unit-Step Response

★ Similarly, by letting $x(nT) = u(nT)$ in the general formula of the time-domain response, the unit-step response of the system can be expressed as

$$y(nT) = \mathcal{R}u(t) = \mathbf{c}^T \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)} \mathbf{b} u(kT) + du(nT)$$

Hence, for $n \geq 0$, we have

$$y(nT) = \mathbf{c}^T \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)} \mathbf{b} + d$$

Example

An initially relaxed system can be represented by the matrices

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ \frac{1}{4} & -\frac{1}{2} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{c}^T = \begin{bmatrix} \frac{7}{8} & \frac{5}{4} \end{bmatrix}, \quad d = \frac{3}{2}$$

Find $h(17T)$.

Solution The impulse response is given by

$$h(nT) = \begin{cases} d_0 & \text{for } n = 0 \\ \mathbf{c}^T \mathbf{A}^{(n-1)} \mathbf{b} & \text{for } n > 0 \end{cases}$$

and hence we have

$$h(17T) = \mathbf{c}^T \mathbf{A}^{16} \mathbf{b}$$

By forming $\mathbf{A}^2$, $\mathbf{A}^4$, and then $\mathbf{A}^{16}$, we get

$$h(17T) = \begin{bmatrix} \frac{7}{8} & \frac{5}{4} \end{bmatrix} \begin{bmatrix} \frac{610}{65{,}536} & -\frac{987}{32{,}768} \\ -\frac{987}{131{,}072} & -\frac{1597}{65{,}536} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1076}{262{,}144} \qquad \blacksquare$$

Advantages

- ★ Discrete-time systems can be analyzed very efficiently through the manipulation of matrices, e.g., by using MATLAB.

Advantages

- Discrete-time systems can be analyzed very efficiently through the manipulation of matrices, e.g., by using MATLAB.
- The state-space representation can be used to characterize and analyze time-dependent systems, (i.e., the elements of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}^T$ could depend on nT).

Advantages

- Discrete-time systems can be analyzed very efficiently through the manipulation of matrices, e.g., by using MATLAB.
- The state-space representation can be used to characterize and analyze time-dependent systems, (i.e., the elements of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}^T$ could depend on nT).
- The state-space representation offers a way for realizing digital filters with increased signal-to-noise ratios.

This slide concludes the presentation.

Thank you for your attention.