Digital Control

CSE421

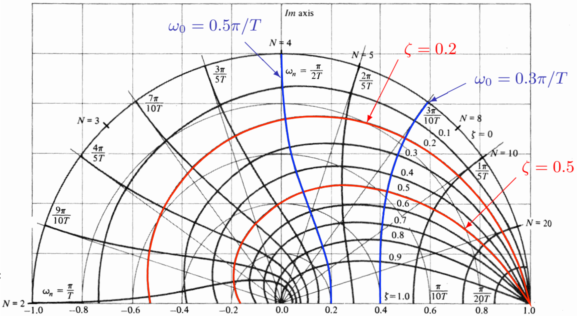
Asst. Prof. Dr.Ing.

Mohammed Nour A. Ahmed

mnahmed@eng.zu.edu.eg

https://mnourgwad.github.io

Lecture 13: **Discrete State-Space Models**



Zagazig University | Faculty of Engineering | Computer and Systems Engineering Department | Zagazig, Egypt

Lecture: 13

Discrete State-Space Models

- State transformation
- Discretization of Continuous-Time State Space Models

Different state-space representations

- As mentioned earlier, there is infinite number of choices for system state, and corresponding system matrices.
- Here, we will introduce the idea of state transformations between different state descriptions.
- Next, **three standard state space descriptions** are presented.

State transformation

- Given state and output equations:

$$\dot{x} = A\,x + B\,u$$
$$y = C\,x + D\,u$$

- let a new state vector z be defined by:

$$z = Tx, \qquad T_{n\times n} \text{ non-singular matrix}$$

- Note that T must be invertible to permit mapping between both state vectors, thus:

$$x = T^{-1}z$$

State transformation

- Now, starting with the new state z, we have:

$$z = Tx$$
$$\dot{z} = T\dot{x} = T(Ax + Bu) = TAx + TBu$$

$$\dot{z} = TAT^{-1}z + TBu \quad \text{new state}$$
$$y = CT^{-1}z + Du \quad \text{description}$$

- the **eigenvalues** of original system matrix A and *new* matrix TAT^{-1} are the same
- For this reason these matrices are called *similar*.
- MATLAB command to transform a state-space model using the transformation $z = Tx$:

```
[A1,B1,C1,D1] = ss2ss(A,B,C,D,T)
```

Controllable Canonical Form

- Consider the third-order transfer function:

$$\frac{Y(s)}{U(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

- To obtain a state space representation of this system, we rewrite:

$$Y(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} U(s) = (b_2 s^2 + b_1 s + b_0) \underbrace{\frac{1}{s^3 + a_2 s^2 + a_1 s + a_0} U(s)}_{V(s)}$$

- The newly introduced variable v is related to u as:

$$\dddot{v} + a_2 \ddot{v} + a_1 \dot{v} + a_0 v = u \quad \Rightarrow \quad \dddot{v} = -a_2 \ddot{v} - a_1 \dot{v} - a_0 v + u$$

- And the output y is formed by:

$$y = b_2 \ddot{v} + b_1 \dot{v} + b_0 v$$

Controllable Canonical Form

- Taking the state variables as:

$$\begin{aligned} x_1 &= v \\ x_2 &= \dot{v} \\ x_3 &= \ddot{v} \end{aligned} \quad \Rightarrow \quad \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -a_0 x_1 - a_1 x_2 - a_2 x_3 + u \end{aligned}$$

- Thus the state space description is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

- This form is known as controllable canonical form. It is a minimal form and is specially useful in finding state-variable feedback laws as we will see later.

Observable Canonical Form

- This form is useful when designing an observer, or state estimator. It can be derived in a similar manner to controllable canonical form, one version of the result is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- With the observable canonical form the output $y = x_1$.

Diagonal (Decoupled) Form

- The simplest representation of SS system is when state equations are decoupled
- that is when system matrix A is diagonal:

$$A = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

- vector x is an **eigenvector**for a matrix A if:

$$A\,x = \lambda\,x$$

- λ is **eigenvalue** (real or complex scalar).
- Eigenvectors represent the **natural response modes** of the system.

Eigenvalues and Eigenvectors

- To find the eigenvalues of a matrix A, we solve the equation:

$$A\,x = \lambda\,x \quad \Rightarrow \quad \lambda x - Ax = 0 \quad \Rightarrow \quad (\lambda I - A)x = 0$$

- For this equation to have a non-trivial solution for x (i.e. $x \neq 0$), matrix $(\lambda I - A)$ must be singular, therefore it must have a zero determinant, i.e.

$$|\lambda I - A| = 0$$

- Note that this equation is also the **characteristic equation** of the system. Therefore, **system poles and eigenvalues are the same** (for minimal state space description!).

Eigenvalues and Eigenvectors

- n^{th} order system will (usually) have n distinct eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$
- Given an eigenvalue λ_i, the corresponding eigenvector x_i is obtained by solving the equation:

$$(\lambda_i I - A)x_i = 0$$

- MATLAB function `eig` finds the eigenvalues and eigenvectors of a matrix A:

```
[Vectors,Values] = eig(A)
```

Example

Find the eigenvalues of the system with the following state space description matrices:

$$A = \begin{bmatrix} 0 & 1 \\ -4 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0$$

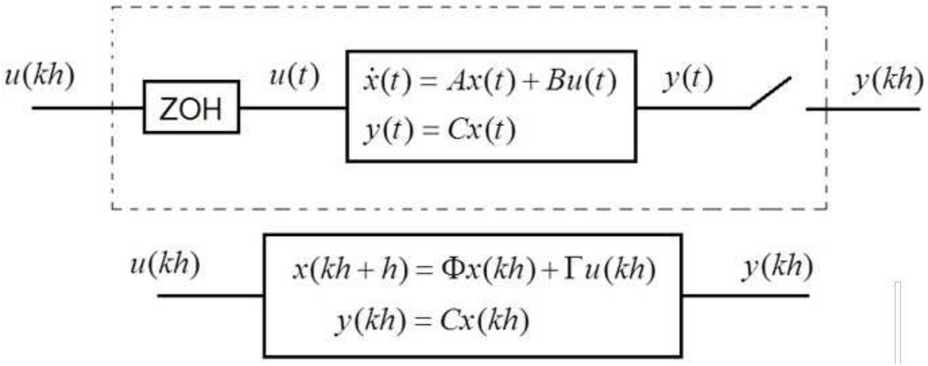
- The eigenvalues are the roots of the following equation:

$$|\lambda I - A| = \begin{vmatrix} \lambda & -1 \\ 4 & \lambda + 2 \end{vmatrix} = \lambda^2 + 2\lambda + 4 = 0 \quad \Rightarrow \lambda_{1,2} = -1 \pm j1.7321$$

Discretization of Continuous-Time State Space Models

- Usually, a physical system is modeled in the form of differential equations (i.e. continuous-time).
- To apply digital control theory, these state space models are transformed to discrete-time (called discretization).

Discrete-time state space model



- To avoid confusion with the continuous system matrices, we use symbols Φ and Γ instead of A and B for the discrete system. h is the sample period.

Solution of the continuous-time state equation

- Consider the state equation:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

- The right-hand side consists of two parts, the first of these, $A\,x(t)$, is the homogeneous part while the second, $B\,u(t)$, is the input portion.
- Let us first examine the homogeneous part.

Solution of the continuous-time state equation

Homogeneous solution

- Consider the homogeneous (no input) state equation

$$\dot{x}(t) = Ax(t)$$

- It can be shown by direct substitution that the solution to this equation is given by

$$x(t) = e^{At}x(0)$$

- where the matrix exponential is defined as

$$e^{At} = I + At + \frac{A^2t^2}{2!} + \frac{A^3t^3}{3!} + \cdots$$

- MATLAB function `expm` can be used for numerical evaluation of the matrix exponential.

Solution of the continuous-time state equation

Homogeneous solution – Another method to evaluate e^{At}

- Taking Laplace transform of:

$$\begin{aligned} \dot{x}(t) &= Ax(t) \\ sX(s) - x(0) &= AX(s), \\ (sI - A)X(s) &= x(0) \\ X(s) &= (sI - A)^{-1}x(0). \end{aligned}$$

- Now, taking the inverse Laplace transform, we get:

$$x(t) = \mathscr{L}^{-1}\left\{(sI - A)^{-1}\right\}\ x(0)$$

- Comparing with the previous solution,

$$x(t) = e^{At}x(0)$$

- gives another way to calculate the matrix exponential:

$$e^{At} = \mathscr{L}^{-1}\{(sI - A)^{-1}\}$$

Special case:

- If the system matrix A is diagonal (the eigenvalues λ_i are along the diagonal), then the matrix exponential is also diagonal and is given by:

$$A = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \Rightarrow e^{At} = \begin{bmatrix} e^{\lambda_1 t} & & & \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ & & & e^{\lambda_n t} \end{bmatrix}$$

The general state equation

- Now consider the general state equation:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

- Taking Laplace transform:

$$\begin{aligned} sX(s) - x(0) &= AX(s) + BU(s), \\ (sI - A)X(s) &= x(0) + BU(s) \\ X(s) &= (sI - A)^{-1}x(0) + (sI - A)^{-1}BU(s) \end{aligned}$$

- Now, taking the inverse Laplace transform, we get:

$$x(t) = e^{At}x(0) + \int_{\tau=0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$$

$$y(t) = Ce^{At}x(0) + C\int_{\tau=0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$$

The general state equation

- Let the initial time be t_0 different from 0, then:

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{\tau=t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$$

- Now define:

$$t_0 = kh, \qquad t = kh + h$$

- Then:

$$x(kh+h) = e^{Ah}x(kh) + \int_{\tau=kh}^{kh+h} e^{A(kh+h-\tau)}Bu(\tau)d\tau$$

- Let us assume, thanks to the ZOH, that:

$$u(t) = u(kh), \quad kh \leq t < kh + h$$

The general state equation

- Then:

$$x(kh+h) = e^{Ah}x(kh) + \left(\int_{\tau=kh}^{kh+h} e^{A(kh+h-\tau)} B d\tau \right) u(kh)$$

- Let us define:

$$\lambda = kh + h - \tau$$

- Then:

$$x(kh+h) = e^{Ah}x(kh) + \left(\int_{\lambda=0}^{h} e^{A\lambda} B \, d\lambda \right) u(kh)$$

The general state equation

- Thus we arrive at the discrete-time state space description:

$$x(kh+h) = \Phi x(kh) + \Gamma u(kh)$$
$$y(kh) = Cx(kh)$$

- Where:

$$\Phi = e^{Ah}, \qquad \Gamma = \int_0^h e^{A\lambda} B d\lambda$$

- Note that the `c2d` command can be used to convert continuous into discrete state space models.

Discretization of Continuous-Time State Space Models

- Given continuous system state-space equations:

$$\dot{x} = A\,x + B\,u$$
$$y = C\,x + D\,u$$

- Its discrete equivalent state-space model is given as:

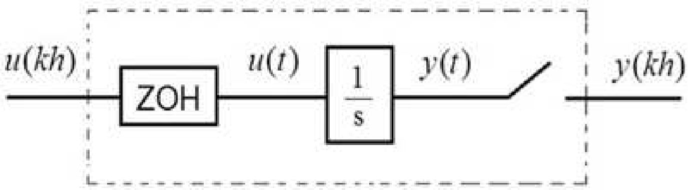
$$x(kh+h) = \Phi x(kh) + \Gamma u(kh)$$
$$y(kh) = Cx(kh)$$

- Where:

$$\Phi = e^{Ah}, \qquad \Gamma = \int_0^h e^{A\lambda}\,B\,d\lambda$$
$$e^{At} = \mathscr{L}^{-1}\{(sI - A)^{-1}\}$$

Example

Find a discrete-time state space description of an integrator, $\dot{y} = u$.



- Let $x_1 = y \quad \Rightarrow \quad \dot{x}_1 = u$
- Then a continuous-time state space model of the system is:

$$\dot{x}_1 = (0)x_1 + (1)u, \qquad y = (1)x_1$$
$$A = 0, \; B = 1, \; C = 1.$$

Discretization of Continuous-Time State Space Models

Example 1

- Then we evaluate the matrices Φ and Γ of the discrete model:

$$\Phi = e^{Ah} = 1, \quad \Gamma = \int_{t=0}^{h} e^{At} B dt = \int_{t=0}^{h} (1) dt = h$$

$$\begin{aligned} x(k+1) &= 1.x(k) + h.u(k) \\ y(k) &= 1.x(k) \end{aligned}$$

Example

Find a discrete-time state space description of a double integrator, $\ddot{y} = u$.

$$x_1 = y \quad \Rightarrow \quad \dot{x}_1 = x_2$$
$$x_2 = \dot{y} \quad \Rightarrow \quad \dot{x}_1 = \ddot{y} = u$$

- Then a continuous-time state space model of the system is:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

Discretization of Continuous-Time State Space Models

Example 2

- The matrix exponential is found as follows:

$$e^{At} = L^{-1}\left\{(sI - A)^{-1}\right\} = L^{-1}\left\{\begin{pmatrix} s & -1 \\ 0 & s \end{pmatrix}^{-1}\right\} = L^{-1}\left\{\frac{1}{s^2}\begin{pmatrix} s & 1 \\ 0 & s \end{pmatrix}\right\}$$

$$= L^{-1}\left\{\begin{pmatrix} 1/s & 1/s^2 \\ 0 & 1/s \end{pmatrix}\right\} = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}$$

- The matrices Φ and Γ are then found as:

$$e^{At} = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \Rightarrow \Phi = e^{Ah} = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$$

$$\Gamma = \int_{t=0}^{h} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} dt = \int_{t=0}^{h} \begin{pmatrix} t \\ 1 \end{pmatrix} dt = \left.\begin{pmatrix} t^2/2 \\ t \end{pmatrix}\right|_0^h = \begin{pmatrix} h^2/2 \\ h \end{pmatrix}$$

- The discrete-time state space model is thus given as:

$$x(kh+h) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x(kh) + \begin{bmatrix} h^2/2 \\ h \end{bmatrix} u(kh)$$

$$y(kh) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(kh)$$

Advantages of state space description

- Well-suited to computer calculation.
- Controller design approach for multiple-inputs multiple-outputs (MIMO) systems is the same as for single-input single-output (SISO) systems.
- With the state-space formulation, the internal behavior of the system is exposed, rather than the input-output modeling of transform (i.e. transfer function based) methods.

Thanks for your attention.

Questions?

Asst. Prof. Dr.Ing.

Mohammed Nour A. Ahmed

mnahmed@eng.zu.edu.eg

`https://mnourgwad.github.io`

Robotics Research Interest Group (zuR^{2}IG)
Zagazig University | Faculty of Engineering | Computer and Systems Engineering Department | Zagazig, Egypt