Lecture 2, Jan 8, 2026

Modelling of Discrete-Time Systems

- LTI systems can be modelled in 3 ways:
  1. *Difference equations* (DEs) of the form $y(k)+a_1y(k-1)+\dots+a_ny(k-n) = b_0u(k)+\dots+b_mu(k-m)$
     - Note we assume that $n \geq m$, which is required for causality (otherwise inputs in the future can affect outputs in the past)
     - This is analogous to ODEs in continuous domain
  2. *Transfer functions* (TFs): $G(z) = \dfrac{Y(z)}{U(z)}$, assuming zero initial conditions
     - They are analogous to transfer functions in continuous time but using a Z-transform instead of the Laplace transform; we also have the ratio of the Z-transform of the output over the input
     - $\dfrac{Y(z)}{U(z)} = \dfrac{b_0 + b_1z^{-1} + \dots + b_mz^{-m}}{1 + a_1z^{-1} + \dots + a_nz^{-n}} = \dfrac{b_0z^n + b_1z^{n-1} + \dots + b_mz^{n-m}}{z^n + a_1z^{n-1} + \dots + a_{n-1}z + a_n} = \dfrac{N(z)}{D(z)}$
  3. State-space models: $x(k+1) = Ax(k) + Bu(k)$ (*canonical form*)
     $$y(k) = Cx(k) + Du(k)$$
     - $x(k) = \begin{bmatrix} x_1(k) \\ \cdots \\ x_n(k) \end{bmatrix} \in \mathbb{R}^n, u(k) = \begin{bmatrix} u_1(k) \\ \cdots \\ u_m(k) \end{bmatrix} \in \mathbb{R}^m, y(k) = \begin{bmatrix} y_1(k) \\ \cdots \\ y_p(k) \end{bmatrix} \in \mathbb{R}^p$

Definition

The *Z-transform* of a discrete time signal $x(k)$ is defined as

$$\mathcal{Z}\{x(k)\} = X(z) = \sum_{k=0}^{\infty} x(k)z^{-k}$$

for some $z \in \mathbb{C}$, provided that the sum is convergent.

- Note the forward and backward shift properties of the Z-transform: $x(k+m)$ transforms to $z^mX(z)$ (for positive or negative m), assuming zero initial conditions
  - Therefore Z-transforming the difference equation we get $Y(z) + a_1z^{-1}Y(z) + \dots + a_nz^{-n}Y(z) = b_0U(z) + \dots + b_mz^{-m}U(z)$, which we can rearrange to get the expression earlier
  - Just like how the Laplace transform converts differentiations to multiplication, the Z-transform converts time shifts to multiplication, turning a difference equation into algebra

Conversion Between State Space and Transfer Function Representations

- To convert from state space to transfer function, take the Z-transform of the state and output equations, and solve for $X(z)$ and $Y(z)$
  - $x(k+1) = Ax(k) + Bu(k) \implies zX(z) = AX(z) + BU(z) \implies X(z) = (zI - A)^{-1}BU(z)$
    * Note $(zI - A)^{-1}$ exists since its form means it can be written as a convergent power series, so it can always be inverted
  - $y(k) = Cx(k) + Du(k) \implies Y(z) = CX(z) + DU(z)$
    $$= C(zI - A)^{-1}BU(z) + DU(z)$$
    $$= (C(zI - A)^{-1}B + D)U(z)$$
  - This gives the transfer function $G(z) = \dfrac{Y(z)}{U(z)} = C(zI - A)^{-1}B + D$
  - Note that we normally only do this for SISO systems (taking the ratio of vectors would not make sense anyway) since transfer functions are generally very hard to work with for MIMO systems
- To convert from transfer function to state space, assume $G(z) = \dfrac{N(z)}{D(z)}$ is a rational proper transfer function, start by converting to a difference equation by inverting the Z-transform, define as many

states as necessary (equal to the order of the difference equation) and organize into matrix form

  - Example: $G(z) = \dfrac{1}{z^2 + a_1 z + a_0}$
  - Convert to difference equation: $$\begin{aligned}&(z^2 + a_1 z + a_0)Y(z) = U(z)\\ \implies\ & y(k+2) + a_1 y(k+1) + a_2 y(k) = u(k)\end{aligned}$$
  - Define states: $x_1(k) = y(k), x_2(k) = y(k+1) \implies \begin{cases} x_1(k+1) = y(k+1) = x_2(k) \\ x_2(k+1) = y(k+2) = u(k) - a_1 x_2(k) - a_2 x_1(k) \end{cases}$
  - $x(k+1) = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$
  - This is known as the *controllable canonical form* (analogous to the same in continuous domain)

- More generally when the numerator is not a constant, rewrite $Y(z) = \dfrac{N(z)}{D(z)} U(z) = N(z)V(z)$ where $V(z) = \dfrac{1}{D(z)} U(z)$, and $V(z)$ becomes the state, $Y(z) = N(z)V(z)$ becomes the output equation, and $V(z) = \dfrac{1}{D(z)} U(z)$ becomes the state equation
  - $G(z) = \dfrac{b_1 z + b_0}{z^3 + a_2 z^2 + a_1 z + a_0} = \dfrac{N(z)}{D(z)}$
  - State equation: $$\begin{aligned}&V(z) = \frac{1}{z^3 + a_2 z^2 + a_1 z + a_0}\\ \implies\ & v(k+3) + a_2 v(k+2) + a_1 v(k+1) + a_0 v(k) = u(k)\end{aligned}$$
    * Define states $x(k) = \begin{bmatrix} v(k) \\ v(k+1) \\ v(k+2) \end{bmatrix} \implies x(k+1) = \begin{bmatrix} x_2(k) \\ x_3(k) \\ -a_2 x_3(k) - a_1 x_2(k) - a_0 x_1(k) + u \end{bmatrix}$
    * Matrix form: $x(k+1) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$
  - Output equation: $$\begin{aligned}&Y(z) = (b_1 z + b_0)V(z)\\ \implies\ & y(k) = b_1 v(k+1) + b_0 v(k)\\ & = b_1 x_2(k) + b_0 x_1(k)\\ & = \begin{bmatrix} b_0 & b_1 & 0 \end{bmatrix} x(k)\end{aligned}$$

- Note that the states obtained by converting from transfer function to state space are not unique
  - Some states are easier to work with than others, so we might want to do a coordinate transform $z(k) = Px(k)$ for some nonsingular P to obtain a new system $z(k+1) = PAP^{-1}z(k) + P^{-1}BU(k)$