Module 02
Control Systems Preliminaries, Intro to State Space

Ahmad F. Taha

EE 5143: Linear Systems and Control

Email: ahmad.taha@utsa.edu

Webpage: http://engineering.utsa.edu/ataha

January 17, 2019

Module 2 Outline

1. Physical laws and equations
2. Transfer function model
3. Model of actual systems
4. Examples
5. From s-domain to time-domain
6. Introduction to state space representation
7. State space canonical forms
8. Analytical examples

Physical Laws and Models

- Any controls course is generally about **dynamical** or **dynamic** systems
- By definition, dynamical systems **are dynamic** because they change with time
- Change in the sense that their intrinsic properties evolve, vary
- Examples: coordinates of a drone, speed of a car, body temperature, concentrations of chemicals in a centrifuge
- Physicists and engineers like to represent dynamic systems with equations—because nerdiness
- Why? Well, the answer is fairly straightforward
- Equations allow us to get away from chaos

Physical Laws

- For many systems, it's easy to understand the physics, and hence the math behind the physics
  - Examples: circuits, motion of a cart, pendulum, suspension system
- For the majority of dynamical systems, the actual physics is complex
- Hence, it can be hard to depict the dynamics with differential eqns
  - Examples: human body temperature, thermodynamics, spacecrafts
- This illustrates the needs for ***models***
- **Dynamic system model:** a mathematical description of the actual physics
- Very important question: **Why do we need a system model?** Because control
- Remember George Box's quote:

ALL MODELS ARE WRONG, BUT SOME ARE USEFUL.

Modeling in Control 101: Transfer Functions?

* **TFs**: *a mathematical representation to describe relationship between inputs and outputs of the physics of a system, i.e., of the differential equations that govern the motion of bodies, for example*

- **Input**: always defined as $u(t)$—called control action
- **Output**: always defined as $y(t)$—called measurement or sensor data
- TF relates the derivatives of $u(t)$ and $y(t)$
- Why is that important? Well, think of $\sum F(t) = ma(t)$
- '$F(t)$' above is the input (exerted forces), and the output is the acceleration, '$a(t)$'

Construction of Transfer Functions

- For linear systems, we can often represent the system dynamics through an nth order ordinary differential equation (ODE):

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + a_{n-2}y^{(n-2)}(t) + \cdots + a_0 y(t) =$$
$$u^{(m)}(t) + b_{m-1}u^{(m-1)}(t) + b_{m-2}u^{(m-2)}(t) + \cdots + b_0 u(t)$$

- The $y^{(k)}$ notation means we're taking the kth derivative of $y(t)$
- Given that ODE description, we can take the Laplace transform (assuming zero initial conditions for all signals)

$$\boxed{\mathcal{L}\left[f^{(n)}(t)\right] = s^n F(s) - s^{n-1}f(0) - s^{n-2}f^{(1)}(0) - \ldots - sf^{(n-2)}(0) - f^{(n-1)}(0)}$$

$$\Rightarrow \boxed{H(s) = \frac{Y(s)}{U(s)} = \frac{s^m + b_{m-1}s^{m-1} + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0}}$$

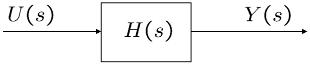
Transfer Functions (Are Boring)

- Given this TF:

$$\boxed{H(s) = \frac{Y(s)}{U(s)} = \frac{s^m + b_{m-1}s^{m-1} + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0}}$$

- For a given control signal $u(t)$ or $U(s)$, we can find the output of the system, $y(t)$, or $Y(s)$
- **Impulse response:** defined as $h(t)$—the output $y(t)$ if the input $u(t) = \delta(t)$
- **Step response:** the output $y(t)$ if the input $u(t) = 1^+(t)$
- For any input $u(t)$, the output is: $\boxed{y(t) = h(t) * u(t)}$
- But...Convolutions are nasty...Who likes them?

TFs of Generic LTI Systems

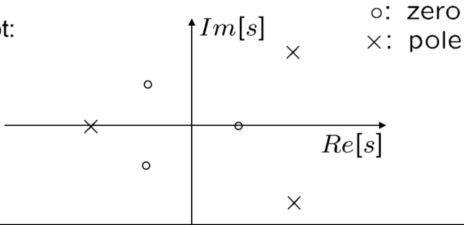


- So, we can take the Laplace transform: $\boxed{Y(s) = H(s)U(s)}$
- Typically, we can write the TF as:

$$\boxed{H(s) = \frac{Y(s)}{U(s)} = \frac{s^m + b_{m-1}s^{m-1} + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0}}$$

- Roots of numerator are called the **zeros** of $H(s)$ or the system
- Roots of the denominator are called the **poles** of $H(s)$

Pole zero plot:



Example

Given: $H(s) = \dfrac{2s+1}{s^3 - 4s^2 + 6s - 4}$

- **Zeros**: $z_1 = -0.5$
- **Poles**: solve $s^3 - 4s^2 + 6s - 4 = 0$, use MATLAB's `roots` command
  * `poles=roots[1 -4 6 -4]`; $poles = \{2, 1+j, 1-j\}$
- **Factored form:**

$$H(s) = 2\frac{s+0.5}{(s-2)(s-1-j)(s-1+j)}$$

Analyzing Generic Physical Systems

Seven-step algorithm:

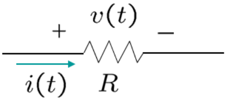
1. Identify dynamic variables, inputs (u), and system outputs (y)
2. Focus on one component, analyze the dynamics (physics) of this component
   - How? Use Newton's Equations, KVL, or thermodynamics laws...
3. After that, obtain an nth order **ODE**:

$$\boxed{\sum_{i=1}^{n} \alpha_i y^{(i)}(t) = \sum_{j=1}^{m} \beta_j u^{(j)}(t)}$$

4. Take the Laplace transform of that **ODE**
5. Combine the equations to eliminate internal variables
6. Write the transfer function from input to output
7. For a certain control $U(s)$, find $Y(s)$, then $y(t) = \mathcal{L}^{-1}[Y(s)]$

Basic Circuits Components

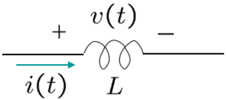
resistor



$$v(t) = Ri(t)$$

$$V(s) = RI(s) \Rightarrow \frac{V(s)}{I(s)} = R$$

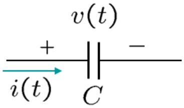
inductor



$$v(t) = L\frac{di(t)}{dt}$$

$$V(s) = LsI(s) \Rightarrow \frac{V(s)}{I(s)} = Ls$$

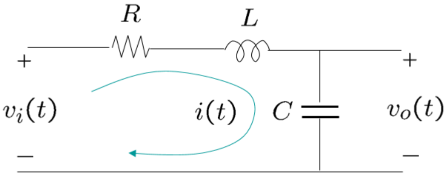
capacitor



$$i(t) = C\frac{dv(t)}{dt}$$

$$I(s) = CsV(s) \Rightarrow \frac{V(s)}{I(s)} = \frac{1}{Cs}$$

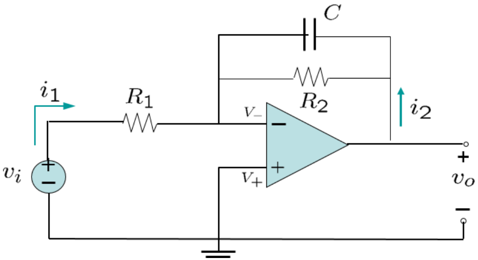
Basic Circuits — RLCs & Op-Amps



$v_i(t)$: input

$v_o(t)$: output

Transfer function $\frac{V_o(s)}{V_i(s)}$

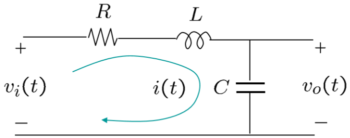


$v_i(t)$: input

$v_o(t)$: output

Transfer function $\frac{V_o(s)}{V_i(s)}$

TF of an RLC Circuit — Example



Objective: Find TF

$v_i(t)$: input

$v_o(t)$: output

Transfer function $\frac{V_o(s)}{V_i(s)}$

- Apply KVL (assume zero ICs):

$$\begin{aligned} v_i(t) &= Ri(t) + L\frac{di(t)}{dt} + \frac{1}{C}\int i(\tau)dt \\ v_o(t) &= \frac{1}{C}\int i(\tau)dt \end{aligned}$$

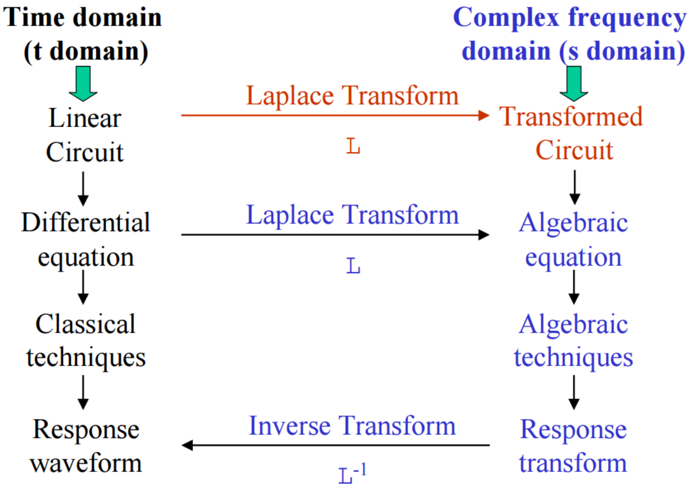
- Take LT for the above differential equations:

$$\begin{aligned} V_i(s) &= RI(s) + LsI(s) + \frac{1}{Cs}I(s) \\ V_o(s) &= \frac{1}{Cs}I(s) \Rightarrow I(s) = CsV_o(s) \end{aligned}$$

$$\Rightarrow \boxed{\frac{V_o(s)}{V_i(s)} = \frac{1}{LCs^2 + RCs + 1}}$$

Generic Circuit Analysis

s-Domain Circuit Analysis



Dynamic Models in Nature

- Predator-prey equations are 1st order non-linear, ODEs
- Describe the dynamics of biological systems where 2 species interact
- One species as a predator and the other as a prey
- Populations change through time according to these equations:

$$\dot{x}(t) = \alpha x(t) - \beta x(t)z(t)$$

$$\dot{z}(t) = \delta x(t)z(t) - \gamma y(t)$$

  - $x(t)$: # of preys (e.g., rabbits)
  - $z(t)$: # of predators (e.g., foxes)
  - $\dot{x}(t), \dot{z}(t)$: growth rates of the 2 species—function of time, t
  - $\alpha, \beta, \gamma, \delta$: $+$ve real parameters depicting the interaction of the species

Mathematical Model

$$\dot{x}(t) = \alpha x(t) - \beta x(t)z(t)$$

$$\dot{z}(t) = \delta x(t)z(t) - \gamma z(t)$$

- Prey's population grows exponentially ($\alpha x(t)$)—why?
- Rate of predation is assumed to be proportional to the rate at which the predators and the prey meet ($\beta x(t)z(t)$)
- If either $x(t)$ or $z(t)$ is zero then there can be no predation
- $\delta x(t)z(t)$ represents the growth of the predator population
- No prey $\Rightarrow$ no food for the predator $\Rightarrow$ $z(t)$ decays
- Is there an equilibrium? What is it?
- No transfer function here

Dynamics in Epidemiology

- **Epidemiology:** The branch of medicine that deals with the incidence, distribution, and possible control of diseases and other factors relating to health
- In the past 10 years, mathematicians, biologists, and physicists studied mathematical models of epidemics
- Why is that important?
- Various models focus on different things:
  - SIR Model: **S** for the number susceptible, **I** for the number of infectious, and **R** for the number recovered
  - SIS Model: Infections like cold and influenza, do not possess lasting immunity
  - SEIR: **E** for exposed
  - MSIR: **M** stands for maternally-derived immunity
  - SEIS and many, many more

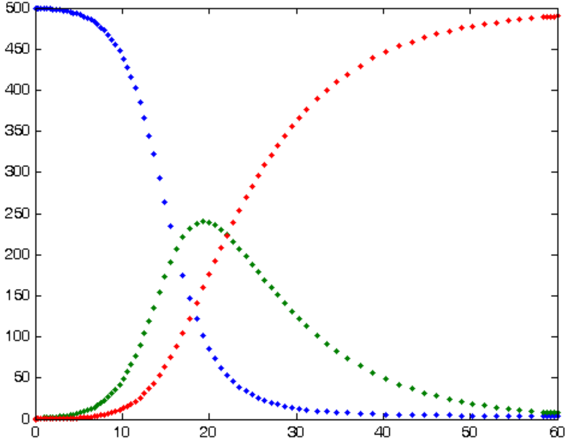
SIR Model

- Here, we present the dynamic model for the SIR model
- We take flu as an example of the SIR model
- Define variable $S(t), I(t), R(t)$ representing the number of people in each category at time t. The SIR model can be written as

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta IS}{N}\\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I\\ \frac{dR}{dt} &= \gamma I.\end{aligned}$$

- N is the total number of people, with $S(t) + I(t) + R(t) = N$
- The force of infection F can be written as $F = \beta I/N$
- β is the contact rate, and γ is the transition rate (rate of recovery)

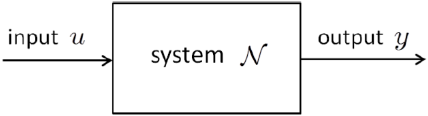
So who do these quantities vary?



Blue represents **Susceptible**, Green represents **Infected**, and Red represents the **Recovered** population.

System Model—Generalization Beyond ODEs

Mathematical model of physical processes:



- System is a signal processor:

$$y = \mathcal{N}(u).$$

- u: input signal
- y: output signal
- $\mathcal{N}$: input-output mapping
- $\mathcal{N}$ could be described by ODEs, PDEs, SDEs, difference equations, algorithms, etc.

Input & Output Signals

- Real vector-valued functions over a time index $\mathcal{I}$:

$$u : \mathcal{I} \to \mathbb{R}^m, \qquad y : \mathcal{I} \to \mathbb{R}^p$$

- Continuous-time signals if $\mathcal{I} = \mathbb{R} = (-\infty, \infty)$:

$$u(t), \ -\infty < t < \infty$$

- Discrete-time signals if $\mathcal{I} = \mathbb{Z} = \{\ldots, -1, 0, 1, \ldots\}$:

$$u[k], \ k = \ldots, -1, 0, 1, \ldots$$

Admissible input set $\mathcal{U}$: set of all input signals u allowed.

- Choice of $\mathcal{U}$ depends on applications
- Example: $u(t) \in \mathcal{U}$ if its Laplace transform $\mathcal{L}[u]$ exists:
  - $u(t)$ is causal: $u(t) = 0, \ \forall t < 0$
  - $u(t)$ is exponentially bounded

Causality in Systems

- **Causality** is the basic property in systems that one process caused another process to happen
- Do not confuse causation with correlation: causation necessitates a relationship between the cause and effect—correlation does not
- Anyway, here's some rigorous definitions

DEF1 A system $\mathcal{N}$ is causal if the output at time t does not depend on the values of the input at any time $t' > t$

DEF2 A system $\mathcal{N}$ mapping x to y is causal IFF for any pair of input signals $x_1(t)$ and $x_2(t)$ such that $x_1(t) = x_2(t), \quad \forall\ t \leq t_0$, the output satisfies

$$y_1(t) = y_2(t), \quad \forall\ t \leq t_0.$$

DEF3 If $h(t)$ is the impulse response of the system $\mathcal{N}$, then the system is causal IFF

$$h(t) = 0, \quad \forall\ t < 0$$

Discrete vs. Continuous & Linear vs. Nonlinear Systems

Discrete-time vs. Continuous-time Systems

System $\mathcal{N}$ is

- a *continuous-time system* if both input and output are continuous-time signals
- a *discrete-time system* if both input and output are discrete-time signals
- a *hybrid system* if both types of signals exist in the system

Linear vs. Nonlinear Systems

System $\mathcal{N}$ is

- a *linear system* if for all $u_1, u_2 \in \mathcal{U}$ and all $\lambda_1, \lambda_2 \in \mathbb{R}$,

$$\mathcal{N}(\lambda_1 u_1 + \lambda_2 u_2) = \lambda_1 \mathcal{N}(u_1) + \lambda_2 \mathcal{N}(u_2)$$

- a *nonlinear system* if otherwise

Time-Invariant vs. Time-Varying & Lumped vs. Distributed Systems

Time-Invariant vs. Time-Varying Systems

System $\mathcal{N}$ is

- a *time-invariant system* if for all $u \in \mathcal{U}$ and all $T \in \mathcal{I}$,

$$y(\cdot) = \mathcal{N}(u(\cdot)) \quad \Rightarrow \quad y(\cdot - T) = \mathcal{N}(u(\cdot - T))$$

- a *time-varying system* if otherwise

Lumped vs. Distributed Systems

System $\mathcal{N}$ is

- a *lumped system* if it has a finite number of state variables
- a *distributed system* if it has an infinite number of state variables

What are the state variables of a system?
State variables is a set of variables whose values at any moment completely characterize the "state-of-the-art" of the system

Examples

Are these systems linear? Nonlinear? TV? TI? Discrete? Continuous? Causal? Non-Causal?

- $y(t) = (u(t))^2$
- $y(t) = t^2 u(t)$
- $y(t) = u(t) - u(t-1)$
- $y(t) = u(t) - u(t+1)$
- $\dot{y}(t) = (u(t))^2 + u(t-1)$
- $y(k+1) = y(k) + u(k)$

Modern Control

- In the undergrad control course, methods that pertain to the analysis and design of control systems via frequency-domain techniques were presented
  - Root locus, PID controllers, compensators, state-feedback control, etc...
  - These studies are considered as the classical control theory—based on the s-domain
- This course focuses on time-domain techniques
  - Theory is based on *State-Space Representations*—modern control
- We will not deal with transfer functions or frequency domain
- Why do we need that? Many reasons

MIMO Systems

- Consider a multi-input, multi-output (MIMO) dynamic system with
  - n **internal states** $(x_1(t), x_2(t), \ldots, x_n(t))$
  - m **control inputs** $(u_1(t), u_2(t), \ldots, u_m(t))$
  - p **measurement outputs** $(y_1(t), y_2(t), \ldots, y_p(t))$
- A scientist comes and gives you the relationship between $x(t), u(t)$, and $y(t)$ as

$$\begin{aligned}
\dot{x}_i(t) &= \sum_{j=1}^{n} a_{ij}x_j(t) + \sum_{j=1}^{m} b_{ij}u_k(t), \;\; \forall i = 1, 2, \ldots, n \\
y_l(t) &= \sum_{j=1}^{n} c_{lj}x_j(t) + \sum_{j=1}^{m} d_{lj}u_j(t), \;\; \forall l = 1, 2, \ldots, p \\
& \quad x_i(0), \;\; \forall i = 1, 2, \ldots, n \;\; \text{are all given}
\end{aligned}$$

  - Constants $a_{ij}, b_{ij}, c_{lj}, d_{lj}$: all given, model parameters of the dynamic system, can be any real number; initial conditions $x_i(0)$ are given
- Almost all dynamic systems can be written as above (we won't learn modeling in this class)

State-Space Representations

- **State-space** (SS) theory: representing the system by a **vector-form first order ODE**:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \quad \boldsymbol{x}_{\text{initial}} = \boldsymbol{x}_{t_0}, \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t), \tag{2}$$

  – $\boldsymbol{x}(t) \in \mathbb{R}^n$: **dynamic state-vector of the LTI system**, $\boldsymbol{u}(t) \in \mathbb{R}^m$: **control input-vector**, $n =$ number of internal states
  – $\boldsymbol{y}(t) \in \mathbb{R}^p$: output-vector and $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ are constant matrices from parameters $a_{ij}, b_{ij}, c_{lj}, d_{lj}$
  – $\boldsymbol{A} \in \mathbb{R}^{n\times n}, \boldsymbol{B} \in \mathbb{R}^{n\times m}, \boldsymbol{C} \in \mathbb{R}^{p\times n}, \boldsymbol{D} \in \mathbb{R}^{p\times m}$
- The above two equations represent a relationship between the input and output of the system via the internal system states
- The above 2 equations are nothing but a matrix-vector first order differential equation
- These equations elegantly model MIMO systems

State-Space to Transfer Functions

- Given a state-space representation:

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t)\end{aligned}$$

can we obtain the transfer function back? **Yes**:

$$\boxed{\frac{Y(s)}{U(s)} = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}}$$

- **Example**: find the TF corresponding for this SISO system:

$$\dot{\boldsymbol{x}}(t) = \underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{x}(t) + \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{\boldsymbol{B}} u(t), \quad y(t) = \underbrace{\begin{bmatrix} 2 & -1 \end{bmatrix}}_{\boldsymbol{C}} x(t) + \underbrace{0}_{D} u(t)$$

- **Solution:**

$$\frac{Y(s)}{U(s)} = \boldsymbol{C}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D} = \begin{bmatrix} 2 & -1 \end{bmatrix} \left(s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0$$

$$= \frac{s+3}{s^2 + 3s + 2}$$

MATLAB Commands

- `ss2tf(A,B,C,D,iu)`
- Demo
- How about obtaining state-space from a transfer function?
- Yes, you can do that too
- `tf2ss(num,den)`

MIMO Systems – Discrete Time

- Previous discussion was on CT systems
- How about discrete time (DT) systems?
- DT systems exist in nature and are also common
- Examples and contrast between DT and CT systems
- Consider a MIMO DT system
  - n **internal states** $(x_1(k), x_2(k), \ldots, x_n(k))$
  - m **control inputs** $(u_1(k), u_2(k), \ldots, u_m(k))$
  - p **measurement outputs** $(y_1(k), y_2(k), \ldots, y_p(k))$
- A scientist comes and gives you the relationship between $x(k), u(k)$, and $y(k)$ as a **difference equation**

$$
\begin{aligned}
x_i(k+1) &= \sum_{j=1}^{n} a_{ij}x_j(k) + \sum_{j=1}^{m} b_{ij}u_k(k), \ \ \forall i = 1, 2, \ldots, n \\
y_l(k) &= \sum_{j=1}^{n} c_{lj}x_j(k) + \sum_{j=1}^{m} d_{lj}u_j(k), \ \ \forall l = 1, 2, \ldots, p \\
& \quad x_i(k=0), \ \ \forall i = 1, \ldots, n \text{ are all given}
\end{aligned}
$$

State-Space Representations

- **State-space** (SS) theory: representing the system by a **vector-form first order difference equation**:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k), \quad \boldsymbol{x}_{\text{initial}} = \boldsymbol{x}_{t_0}, & (3) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k), & (4) \end{aligned}$$

  - $\boldsymbol{x}(k) \in \mathbb{R}^n$: **dynamic state-vector of the LTI system**, $\boldsymbol{u}(k) \in \mathbb{R}^m$: **control input-vector**, $n =$ number of internal states
  - $\boldsymbol{y}(k) \in \mathbb{R}^p$: output-vector and $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ are constant matrices from parameters $a_{ij}, b_{ij}, c_{lj}, d_{lj}$
  - $\boldsymbol{A} \in \mathbb{R}^{n\times n}, \boldsymbol{B} \in \mathbb{R}^{n\times m}, \boldsymbol{C} \in \mathbb{R}^{p\times n}, \boldsymbol{D} \in \mathbb{R}^{p\times m}$

Important Remarks

- So why do we want to go from a transfer function to a time-representation, ODE form of the system?
- There are many benefits for doing so, such as:
  1. Stability analysis for MIMO systems becomes way easier
  2. We have powerful mathematical tools that help us design controllers
  3. RL and compensator designs were relatively tedious design problems
  4. With state-space representations, we can easily design controllers
  5. Nonlinear dynamics: cannot use TFs for nonlinear systems
  6. State-space is all about time-domain analysis, which is far more intuitive than frequency-domain analysis
  7. With Laplace transforms and TFs, we had to take inverse Laplace transforms. In many cases, the Laplace transform does not exist, which means time-domain analysis is the only way to go
- We will learn how to get a solution for $y(t)$ for any given $u(t)$ from the state-space representation of the system without Laplace transform—via ODE solutions for matrix-vector equations

State Space Generalization: Nonlinear Systems

A continuous-time lumped system with the state $x(t) \in \mathbb{R}^n$:

$$\begin{cases} \frac{dx}{dt} = f\left(x(t), u(t), t\right) \\ y(t) = g\left(x(t), u(t), t\right) \end{cases}, \quad -\infty < t < \infty$$

- $x(t) \in \mathbb{R}^n$: state
- $u(t) \in \mathbb{R}^m$: input
- $y(t) \in \mathbb{R}^p$: output

A discrete-time lumped system with the state $x[k] \in \mathbb{R}^n$:

$$\begin{cases} x[k+1] = f\left(x[k], u[k], k\right) \\ y[k] = g\left(x[k], u[k], k\right) \end{cases}, \quad k = \ldots, -1, 0, 1, \ldots$$

- $x[k] \in \mathbb{R}^n$: state
- $u[k] \in \mathbb{R}^m$: input
- $y[k] \in \mathbb{R}^p$: output

State Space Generalization: LTV Systems

A continuous-time lumped linear system with state $x(t) \in \mathbb{R}^n$:

$$\begin{cases} \frac{dx}{dt} = A(t)x(t) + B(t)u(t) \\ y(t) = C(t)x(t) + D(t)u(t) \end{cases}, \quad -\infty < t < \infty$$

where $A(t), B(t), C(t), D(t)$ are matrices of proper dimension

A discrete-time lumped linear system with state $x[k] \in \mathbb{R}^n$:

$$\begin{cases} x[k+1] = A[k]x[k] + B[k]u[k] \\ y[k] = C[k]x[k] + D[k]u[k] \end{cases}, \quad k = \ldots, -1, 0, 1, \ldots$$

where $A[k], B[k], C[k], D[k]$ are matrices of proper dimension

State Space Generalization: LTI Systems

A continuous-time lumped LTI system with state $x(t) \in \mathbb{R}^n$:

$$\begin{cases} \frac{dx}{dt} = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}, \quad -\infty < t < \infty$$

where A, B, C, D are constant matrices of proper dimension

A discrete-time lumped linear system with state $x[k] \in \mathbb{R}^n$:

$$\begin{cases} x[k+1] = Ax[k] + Bu[k] \\ y[k] = Cx[k] + Du[k] \end{cases}, \quad k = \ldots, -1, 0, 1, \ldots$$

where A, B, C, D are constant matrices of proper dimension

Important Remarks, Milestones

- We have introduced state-space (SS) representations
- The main use of SS is to generate real-time values and numerical solutions for $\boldsymbol{x}(t)$, the vector that includes the states of the system
- The main problem to be solved here is: *Given an initial condition for system $\boldsymbol{x}(0)$ and a control input $\boldsymbol{u}(t)$ (single input (scalar), or multiple inputs (vector)), what will the state of the system ($\boldsymbol{x}(t)$) be? What about $\boldsymbol{y}(t)$?*
- To answer this question, we need to find a solution to the *matrix-vector differential equation*:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$$

- If the system has one state, no controls, the solution is obvious
- If the system has multiple states, controls, solution is a bit complicated
- To find the answer to the above question, we will have to go through a review of basic mathematical concepts—next Module

Questions And Suggestions?

Thank You!

Please visit

engineering.utsa.edu/ataha

IFF you want to know more ☺