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/~taha

ELECTRICAL & COMPUTER ENGINEERING
UTSA ROADRUNNERS

August 28, 2017

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 $\sum F = ma$
- 'F' above is the input (exerted forces), and the output is the acceleration, 'a'

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)

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

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

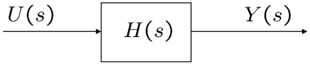
Transfer Functions (Are Boring)

- Given this TF:

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

- 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

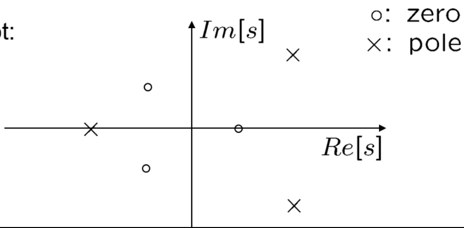


- So, we can take the Laplace transform: $\boxed{Y(s) = H(s)U(s)}$
- Typically, we can write the TF as:

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

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

- **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:**

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

- Please go through http://engineering.utsa.edu/~taha/teaching2/EE3413_Module2.pdf for a review of Laplace transforms and ODEs

Analyzing Generic Physical Systems

Seven-step algorithm:

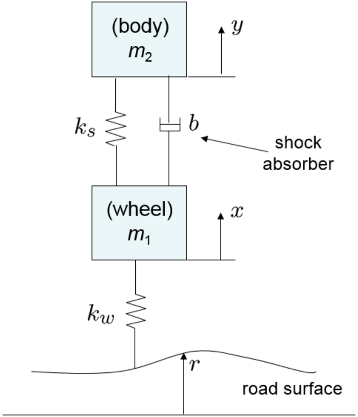
1. Identify dynamic variables, inputs (u), and system outputs (y)
2. Focus on one component, analyze the dynamics (physics) of this component
   - How? Use Newton's Equations, KVL, or thermodynamics laws...
3. After that, obtain an nth order **ODE**:

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

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 $y(t) = \mathcal{L}^{-1}[Y(s)]$

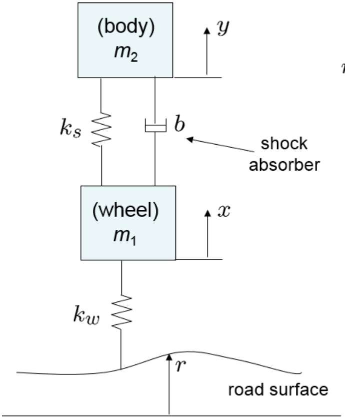
Active Suspension Model

- Each car has 4 active suspension systems (on each wheel)
- System is nonlinear, but we consider approximation. **Objective?**
- **Input**: road altitude $r(t)$ (or $u(t)$), **Output**: car body height $y(t)$



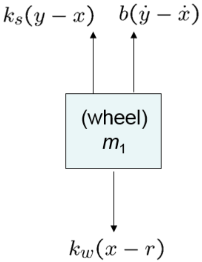
Active Suspension Model — Equations for 1 Wheel

- We only consider one of the four systems
- System has many components, most important ones are: body (m_2) & wheel (m_1) weights

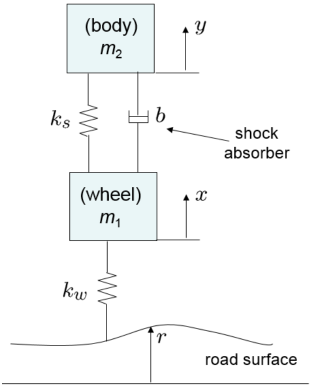


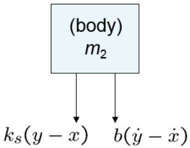
By Newton's Second Law

$$m_1\ddot{x} = k_s(y-x) + b(\dot{y}-\dot{x}) - k_w(x-r)$$



Active Suspension Model — Equations for Car Body





By Newton's Second Law

$$m_2\ddot{y} = -k_s(y-x) - b(\dot{y}-\dot{x})$$

- We now have 2 equations depicting the car body and wheel motion
- Objective: find the TF relating output $(y(t))$ to input $(r(t))$
- What is $H(s) = \dfrac{Y(s)}{R(s)}$?

Active Suspension Model — Transfer Function

- **Differential equations (in time):**

$$\begin{aligned} m_1\ddot{x}(t) &= k_s(y(t)-x(t))+b(\dot{y}(t)-\dot{x}(t))-k_w(x(t)-r(t)) \\ m_2\ddot{y}(t) &= -k_s(y(t)-x(t))-b(\dot{y}(t)-\dot{x}(t)) \end{aligned}$$

- **Take Laplace transform given zero ICs:**

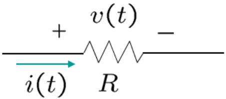
– **Solution:**

- Find $H(s) = \dfrac{Y(s)}{R(s)}$

– **Solution:**

Basic Circuits Components

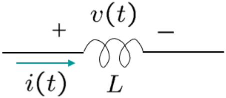
resistor



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

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

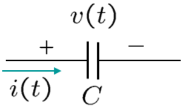
inductor



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

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

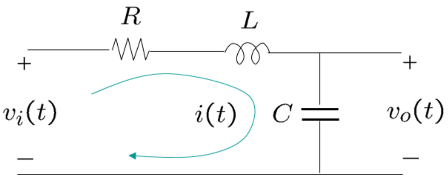
capacitor



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

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

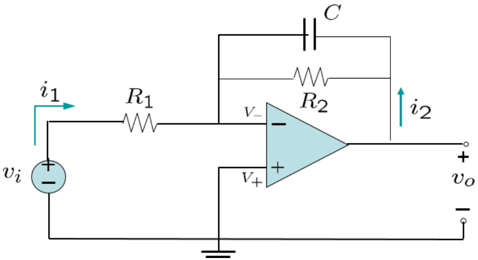
Basic Circuits — RLCs & Op-Amps



$v_i(t)$: input

$v_o(t)$: output

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

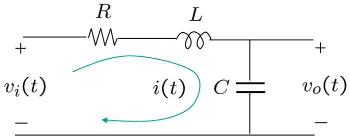


$v_i(t)$: input

$v_o(t)$: 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):

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

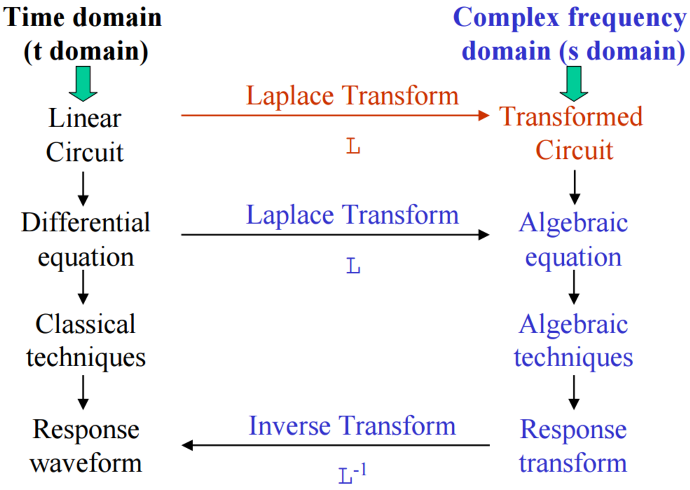
- Take LT for the above differential equations:

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

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

Generic Circuit Analysis

s-Domain Circuit Analysis



General Discussion on Equivalent Systems

Translational Mechanical	Rotational Mechanical	Series RLC Circuit	Parallel RLC Circuit
Position x	Angle θ	Charge q	Flux linkage ϕ
Velocity $\frac{dx}{dt}$	Angular velocity $\frac{d\theta}{dt}$	Current $\frac{dq}{dt}$	Voltage $\frac{d\phi}{dt}$
Mass M	Moment of inertia I	Inductance L	Capacitance C
Spring constant K	Torsion constant μ	Elastance $1/C$	Magnetic reluctance $1/L$
Damping ζ	Rotational friction Γ	Resistance R	Conductance $G = 1/R$
Drive force $F(t)$	Drive torque $\tau(t)$	Voltage e	Current i
Undamped resonant frequency f_n :			
$\frac{1}{2\pi}\sqrt{\frac{K}{M}}$	$\frac{1}{2\pi}\sqrt{\frac{\mu}{I}}$	$\frac{1}{2\pi}\sqrt{\frac{1}{LC}}$	$\frac{1}{2\pi}\sqrt{\frac{1}{LC}}$
Differential equation:			
$M\ddot{x} + \zeta\dot{x} + Kx = F$	$I\ddot{\theta} + \Gamma\dot{\theta} + \mu\theta = \tau$	$L\ddot{q} + R\dot{q} + q/C = e$	$C\ddot{\phi} + G\dot{\phi} + \phi/L = i$

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:

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

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

  - $x(t)$: # of preys (e.g., rabbits)
  - $y(t)$: # of predators (e.g., foxes)
  - $\dot{x}(t), \dot{y}(t)$: 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

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

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

- 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 ($\beta x(t)y(t)$)
- If either $x(t)$ or $y(t)$ is zero then there can be no predation
- $\delta x(t)y(t)$ represents the growth of the predator population
- No prey $\Rightarrow$ no food for the predator $\Rightarrow$ $y(t)$ decays
- Is there an equilibrium? What is it?

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

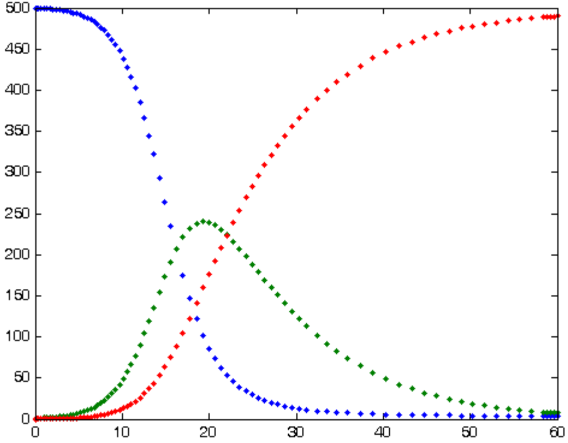
SIR Model

- 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)

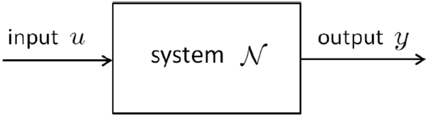
So who do these quantities vary?



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
- Why do we need that? Many reasons

ODEs & Transfer Functions

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

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + a_2 y^{(n-2)}(t) + \cdots + a_{n-1}\dot{y}(t) + a_n y(t) =$$

$$b_0 u^{(n)}(t) + b_1 u^{(n-1)}(t) + b_2 u^{(b-2)}(t) + \cdots + b_{n-1}\dot{u}(t) + b_n u(t)$$

- Input: $u(t)$; Output: $y(t)$—What if we have MIMO system?
- Given that ODE description, we can take the LT (assuming zero initial conditions for all signals):

$$\boxed{H(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1}s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n}}$$

ODEs & TFs

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1}s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n}$$

- This equation represents relationship between **one system input and one system output**
- This relationship, however, **does not show me the internal states of the system, nor does it explain the case with multi-input system**
- For that (and other reasons), we discuss the notion of **system state**
- **Definition:** $\boldsymbol{x}(t)$ is a state-vector that belongs to $\mathbb{R}^n$: $\boldsymbol{x}(t) \in \mathbb{R}^n$
- $\boldsymbol{x}(t)$ is an internal state of a system
- Examples: voltages and currents of circuit components

ODEs, TFs to State-Space Representations

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1}s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n}$$

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

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \quad \boldsymbol{x}_{\text{initial}} = \boldsymbol{x}_{t_0}, &(1)\\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t), &(2) \end{aligned}$$

  - $\boldsymbol{x}(t) \in \mathbb{R}^n$: **dynamic state-vector of the LTI system**, $\boldsymbol{u}(t)$: **control input-vector**, $n =$ order of the TF/ODE
  - $\boldsymbol{y}(t)$: output-vector and $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ are constant matrices
  - For the above transfer function, we have one input $U(s)$ and one output $Y(s)$, hence the size of $y(t)$ and $u(t)$ is only one (scalars), while the size of vector $\boldsymbol{x}(t)$ is n, which is the order of the TF

- **Objective:** learn how to construct matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ given a TF

State-Space Representation 1

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n}$$

- Given the above TF/ODE, we want to find

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

- 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 first order differential equation
- Wait, WHAT? But the TF/ODE was an nth order ODE. How do we have a **first order ODE** now?
- Well, because this equation is vector-matrix equation, whereas the ODE/TF was a scalar equation
- Next, we'll learn how to get to these 2 equations from any TF

State-Space Representation 2 [Ogata, P. 689]

$$\frac{Y(s)}{U(s)} = b_0 + \frac{(b_1 - a_1 b_0)s^{n-1} + \cdots + (b_{n-1} - a_{n-1}b_0)s + (b_n - a_n b_0)}{s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n}$$

which can be modified to

$$Y(s) = b_0 U(s) + \hat{Y}(s) \tag{9–71}$$

where

$$\hat{Y}(s) = \frac{(b_1 - a_1 b_0)s^{n-1} + \cdots + (b_{n-1} - a_{n-1}b_0)s + (b_n - a_n b_0)}{s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n} U(s)$$

Let us rewrite this last equation in the following form:

$$\frac{\hat{Y}(s)}{(b_1 - a_1 b_0)s^{n-1} + \cdots + (b_{n-1} - a_{n-1}b_0)s + (b_n - a_n b_0)} = \frac{U(s)}{s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n} = Q(s)$$

From this last equation, the following two equations may be obtained:

$$s^n Q(s) = -a_1 s^{n-1} Q(s) - \cdots - a_{n-1} s Q(s) - a_n Q(s) + U(s) \tag{9–72}$$

$$\hat{Y}(s) = (b_1 - a_1 b_0)s^{n-1} Q(s) + \cdots + (b_{n-1} - a_{n-1}b_0)sQ(s) + (b_n - a_n b_0)Q(s) \tag{9–73}$$

State-Space Representation 3 [Ogata, P. 689]

Now define state variables as follows:

$$
\begin{aligned}
X_1(s) &= Q(s) \\
X_2(s) &= sQ(s) \\
&\vdots \\
X_{n-1}(s) &= s^{n-2}Q(s) \\
X_n(s) &= s^{n-1}Q(s)
\end{aligned}
$$

Then, clearly,

$$
\begin{aligned}
sX_1(s) &= X_2(s) \\
sX_2(s) &= X_3(s) \\
&\vdots \\
sX_{n-1}(s) &= X_n(s)
\end{aligned}
$$

State-Space Representation 4 [Ogata, P. 689]

which may be rewritten as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \end{aligned} \qquad (9\text{–}74)$$

Noting that $s^n Q(s) = sX_n(s)$, we can rewrite Equation (9–72) as

$$sX_n(s) = -a_1 X_n(s) - \cdots - a_{n-1}X_2(s) - a_n X_1(s) + U(s)$$

or

$$\dot{x}_n = -a_n x_1 - a_{n-1}x_2 - \cdots - a_1 x_n + u \qquad (9\text{–}75)$$

Also, from Equations (9–71) and (9–73), we obtain

$$\begin{aligned} Y(s) &= b_0 U(s) + (b_1 - a_1 b_0)s^{n-1}Q(s) + \cdots + (b_{n-1} - a_{n-1}b_0)sQ(s) \\ &\quad + (b_n - a_n b_0)Q(s) \\ &= b_0 U(s) + (b_1 - a_1 b_0)X_n(s) + \cdots + (b_{n-1} - a_{n-1}b_0)X_2(s) \\ &\quad + (b_n - a_n b_0)X_1(s) \end{aligned}$$

The inverse Laplace transform of this output equation becomes

$$y = (b_n - a_n b_0)x_1 + (b_{n-1} - a_{n-1}b_0)x_2 + \cdots + (b_1 - a_1 b_0)x_n + b_0 u \qquad (9\text{–}76)$$

Final Solution

- Combining equations (9-74,75,76), we can obtain the following **vector-matrix first order differential equation:**

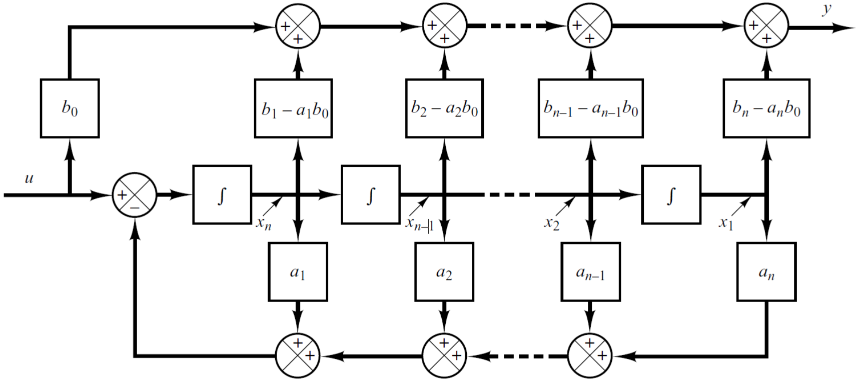
$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix}}_{\boldsymbol{Ax}(t)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t)}_{\boldsymbol{B}u(t)}$$

$$y(t) = \underbrace{\begin{bmatrix} b_n - a_n b_0 | & b_{n-1} - a_{n-1} b_0 | & \cdots | & b_1 - a_1 b_0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix}}_{\boldsymbol{Cx}(t)} + \underbrace{b_0 u(t)}_{Du(t)}$$

Remarks

- For any TF with order n (order of the denominator), with one input and one output:
  - $\boldsymbol{A} \in \mathbb{R}^{n \times n}, \boldsymbol{B} \in \mathbb{R}^{n \times 1}, \boldsymbol{C} \in \mathbb{R}^{1 \times n}, \boldsymbol{D} \in \mathbb{R}$
  - Above matrices are constant $\Rightarrow$ system is linear **time-invariant** (LTI)
  - If one term of the TF/ODE (i.e., the a's and b's) change as a function of time, the matrices derived above will also change in time $\Rightarrow$ system is linear **time-varying** (LTV)
- The above state-space form is called the *controllable canonical form*
- You can come up with different forms of $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ matrices given a different transformation

State-Space and Block Diagrams



- From the derived eqs. before, you can construct the block diagram
- An integrator block is equivalent to a $\frac{1}{s}$, the inputs and outputs of each integrator are the derivative of the state $\dot{x}_i(t)$ and $x_i(t)$
- A system (TF/ODE) of order n can be constructed with n integrators (you can construct the system with more integrators)

Example 1

- Find a state-space representation (i.e., the state-space matrices) for the system represented by this second order transfer function:

$$\frac{Y(s)}{U(s)} = \frac{s+3}{s^2+3s+2}$$

- **Solution:** look at the previous slides with the matrices:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1}s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n} = \frac{\overbrace{0}^{b_0} s^2 + \overbrace{1}^{b_1} s + \overbrace{3}^{b_2}}{s^2 + \underbrace{3}_{a_1} s + \underbrace{2}_{a_2}}$$

  - First, $n = 2 \Rightarrow \boldsymbol{A} \in \mathbb{R}^{2\times 2}, \boldsymbol{B} \in \mathbb{R}^{2\times 1}, \boldsymbol{C} \in \mathbb{R}^{1\times 2}, \boldsymbol{D} \in \mathbb{R}$

$$\dot{\boldsymbol{x}}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{x}(t) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\boldsymbol{B}} u(t)$$

$$y(t) = \underbrace{\begin{bmatrix} 3 & 1 \end{bmatrix}}_{\boldsymbol{C}} x(t) + \underbrace{0}_{D} u(t)$$

Other State-Space Forms Given a TF/ODE[1]

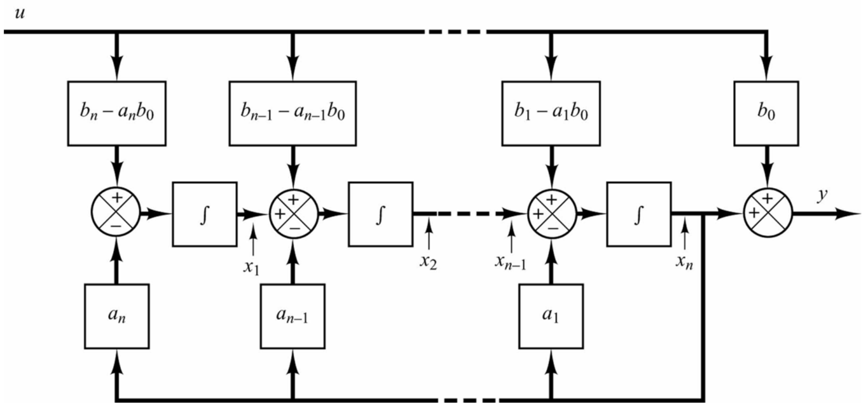
Observable Canonical Form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_n \\ 1 & 0 & \cdots & 0 & -a_{n-1} \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & \cdots & 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + \begin{bmatrix} b_n - a_n b_0 \\ b_{n-1} - a_{n-1} b_0 \\ \cdot \\ \cdot \\ \cdot \\ b_1 - a_1 b_0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \end{bmatrix} + b_0 u$$

[1] Derivation from Ogata, but similar to the controllable canonical form.

Block Diagram of Observable Canonical Form



Other State-Space Forms Given a TF/ODE

Diagonal Canonical Form[2]*:*

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1} s + b_n}{(s+p_1)(s+p_2)\cdots(s+p_n)}$$

$$= b_0 + \frac{c_1}{s+p_1} + \frac{c_2}{s+p_2} + \cdots + \frac{c_n}{s+p_n}$$

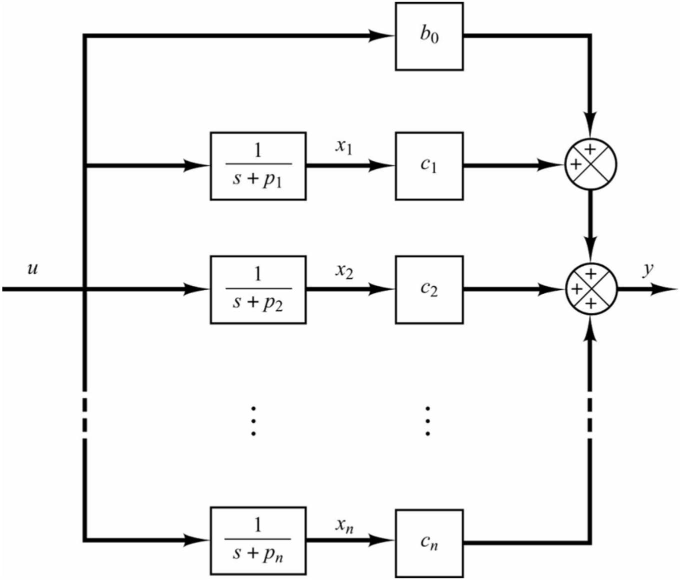
$$\Downarrow \qquad \Downarrow \qquad \Downarrow$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} -p_1 & & & & & 0 \\ & -p_2 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ 0 & & & & & -p_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + b_0 u$$

[2] This factorization assumes that the TF has only distinct real poles.

Block Diagram of Diagonal Canonical Form



Example 1 Solution for other Canonical Forms

- Find the observable and diagonal forms for

$$\frac{Y(s)}{U(s)} = \frac{\overbrace{0}^{b_0} s^2 + \overbrace{1}^{b_1} s + \overbrace{3}^{b_2}}{s^2 + \underbrace{3}_{a_1} s + \underbrace{2}_{a_2}}$$

- **Solution:** look at the previous slides with the constructed state-space matrices:

– **Observable Canonical Form:**

$$\dot{\boldsymbol{x}}(t) = \underbrace{\begin{bmatrix} 0 & -2 \\ 1 & -3 \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{x}(t) + \underbrace{\begin{bmatrix} 3 \\ 1 \end{bmatrix}}_{\boldsymbol{B}} u(t), \;\; y(t) = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{\boldsymbol{C}} x(t) + \underbrace{0}_{D} u(t)$$

– **Diagonal Canonical Form:**

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

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:**

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

$$= \frac{s+3}{s^2+3s+2},$$ that's the TF from the previous example!

MATLAB Commands

- `ss2tf(A,B,C,D,iu)`
- `tf2ss(num,den)`
- Demo

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 Lumped 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/~taha

IFF you want to know more ☺