Introduction to Linear Dynamical Systems

Engineering Examples and State-Space Conversion

Dr. Yu Zhang

Department of Electrical and Computer Engineering
University of California, Santa Cruz

Fall 2025

Engineering Examples of Linear Dynamical Systems

Linear dynamical systems arise in many fields of engineering and science:

- Electrical circuits (RLC networks, filters)
- Mechanical systems (mass-spring-damper, suspension)
- Aerospace (aircraft stability, flight control)
- Robotics (joint and manipulator dynamics)
- Communications (filters, channel models)
- Power systems (generator swing dynamics, grid stability)
- Economics (linearized market and supply-demand models)

General State-Space Model

Continuous-Time State-Space Equations:

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t) + Du(t)$$

Definitions:

- $x(t) \in \mathbb{R}^n$: State vector (internal variables that capture system dynamics)
- $u(t) \in \mathbb{R}^m$: Input vector (external signals that drive the system)
- $y(t) \in \mathbb{R}^p$: Output vector (measured or controlled variables)

The Matrices in State-Space Form

System Matrix $A \in \mathbb{R}^{n \times n}$

- Governs how the state evolves on its own (without input)
- Encodes natural dynamics, eigenvalues, and stability
- Example: In a mass–spring–damper system, A determines natural frequency and damping
- *Intuition:* A tells you "if I leave the system alone, how do states evolve?"

Input Matrix $B \in \mathbb{R}^{n \times m}$

- Maps the external input $u(t)$ into the state equations
- Determines how each input affects each state
- *Intuition:* B tells you "when I push on the system, which states move and how strongly?"

The Matrices in State-Space Form (cont.)

Output Matrix $C \in \mathbb{R}^{p \times n}$

- Maps the internal states to the measurable outputs
- Determines what part of the state vector is visible at the output
- *Intuition:* C tells you "which states am I actually measuring or observing?"

Feedthrough/Direct Matrix $D \in \mathbb{R}^{p \times m}$

- Maps the inputs directly to the outputs (without going through the state)
- $D = 0$ in many physical systems (no instantaneous effect), but not always
- Example: In resistor circuits, input voltage can instantly affect output current
- *Intuition:* D tells you "does the input show up instantly in the output, bypassing states?"

Example: Electrical Circuit (RLC)

System: Series RLC circuit driven by voltage $u(t)$.

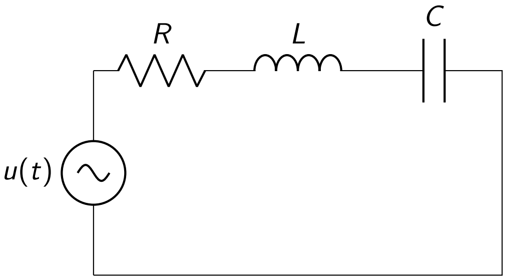
Governing equations:

$$Ri(t) + L\frac{di(t)}{dt} + u_c(t) = u(t)$$

state-space form:

$$\begin{aligned}\dot{x}_1 &= x_2,\\ \dot{x}_2 &= -\tfrac{R}{L}x_2 - \tfrac{1}{LC}x_1 + \tfrac{1}{L}u(t),\end{aligned}$$

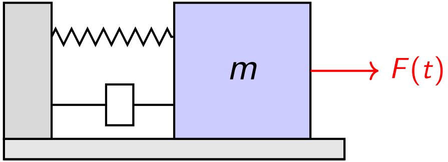
where $x_1 = Cu_c(t)$, $x_2 = i(t)$.



Linear ODE system from KVL/KCL.

Mechanical System: Mass–Spring–Damper

- Block of mass m attached to wall with spring (stiffness k) and damper (damping c).
- The spring (like a rubber band) wants to restore position. The damper (shock absorber) resists motion and kills vibrations, which **improves stability and comfort**
- External force $F(t)$ applied to the block. Displacement from equilibrium: $x(t)$.



Models vibrations and suspension dynamics.

The damper provides a force proportional to velocity that opposes motion:

- **Dissipates energy** $\rightarrow$ prevents endless oscillations.
- **Controls response type:**
  - Underdamped ($c < c_{\text{crit}}$): oscillatory decay.
  - Critically damped ($c = c_{\text{crit}}$): fastest return to equilibrium without oscillation.
  - Overdamped ($c > c_{\text{crit}}$): no oscillation, but sluggish return.

Governing Equation

Newton's Second Law:

$$m\ddot{x}(t) = F_{\text{spring}} + F_{\text{damper}} + F_{\text{external}}.$$

- Spring force: $F_{\text{spring}} = -kx(t)$ (Hooke's Law).
- Damper force: $F_{\text{damper}} = -c\dot{x}(t)$.
- External force: $F_{\text{external}} = F(t)$.

Final governing ODE:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t).$$

Why a Linear Dynamical System?

- The equation is **linear** in $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$.
- It is **dynamical**: the future state depends on displacement, velocity, and the input force.
- Compact representation possible using state-space form.

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t)$$

State-Space Representation

Define states:

$$x_1 = x(t), \quad x_2 = \dot{x}(t).$$

Then:

$$\dot{x}_1 = x_2, \qquad \dot{x}_2 = -\frac{k}{m}x_1 - \frac{c}{m}x_2 + \frac{1}{m}F(t).$$

state-space matrices:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = [0].$$

Physical Interpretation

- **Matrix** A**:** Intrinsic system dynamics (interaction of displacement and velocity).
- **Matrix** B**:** How external input force $F(t)$ affects the system.
- **Matrix** C**:** Chooses displacement $x(t)$ as output.
- **Matrix** D**:** Zero (input does not appear directly in output).

$$\dot{x} = Ax + Bu, \quad y = Cx + Du$$

Engineering Applications

The mass–spring–damper model underlies many real systems:

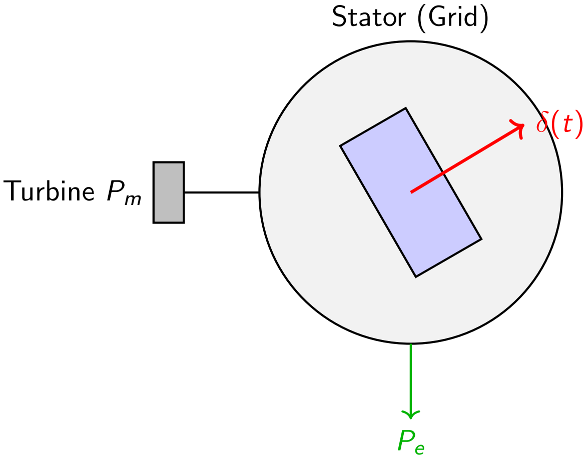
- **Automotive suspensions:** ride comfort and vibration isolation.
- **Civil engineering:** building dynamics under wind or earthquake loads.
- **Robotics/aerospace:** vibration damping in structures and arms.
- **Control theory:** canonical test system for PID and state-feedback design.

Power Grid Example: Swing Equation (Generator Rotor Dynamics)

- Rotor angle $\delta(t)$ is measured relative to the synchronous reference.
- First derivative: $\dot{\delta}(t) = \Delta\omega(t)$ is speed deviation.
- Second derivative follows from the **Swing Equation**:

$$M\,\ddot{\delta}(t) + D\,\dot{\delta}(t) \;=\; P_m(t) - P_e(\delta(t)),$$

  where:
  - M inertia, resists acceleration.
  - D damping, resists oscillations.
  - P_m mechanical input power, P_e electrical output power.



Operating Point and Small Deviations

- Choose an operating point $(\delta_0,\ \dot{\delta} = 0)$ with power balance

$$P_{m0} \;=\; P_e(\delta_0).$$

- Define small deviations:

$$\Delta\delta = \delta - \delta_0, \quad \Delta\omega = \dot{\delta}, \quad \Delta P_m = P_m - P_{m0}.$$

- Linearize electrical power around δ_0:

$$P_e(\delta) \;\approx\; P_e(\delta_0) \;+\; K_s\,\Delta\delta, \qquad K_s \;=\; \left.\frac{dP_e}{d\delta}\right|_{\delta_0}.$$

Linearization around the Operating Point

Start from

$$M\,\ddot{\delta} + D\,\dot{\delta} \;=\; P_m - P_e(\delta).$$

Subtract the equilibrium and use the linearization:

$$M\,\Delta\ddot{\delta} + D\,\Delta\dot{\delta} \;=\; \Delta P_m \;-\; K_s\,\Delta\delta.$$

State-space form with $x = \begin{bmatrix} \Delta\delta \\ \Delta\omega \end{bmatrix}$, $\Delta\omega = \Delta\dot{\delta}$, $u = \Delta P_m$:

$$\begin{aligned} \Delta\dot{\delta} &= \Delta\omega, \\ M\,\Delta\dot{\omega} &= \Delta P_m - K_s\,\Delta\delta - D\,\Delta\omega, \end{aligned} \qquad \Longrightarrow \qquad \dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{K_s}{M} & -\frac{D}{M} \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix}}_{B} u.$$

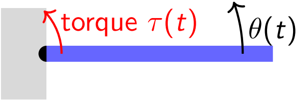
Robotics Example: Single-Joint Position Control

Joint Dynamics:

$$J\ddot{\theta}(t) + b\dot{\theta}(t) = \tau(t).$$

State-space:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{J} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \tau(t), \qquad x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$$



Interpretation:

- Models one rotary robotic joint.
- J: inertia $\rightarrow$ resists acceleration.
- b: damping $\rightarrow$ frictional losses.
- τ: motor torque (control input).

From ODE to State-Space (1/2)

General idea: Any higher-order ODE can be rewritten as a first-order system.

General nth-order ODE:

$$a_n y^{(n)} + a_{n-1} y^{(n-1)} + \cdots + a_1 \dot{y} + a_0 y = b_0 u(t).$$

Step 1. Define the states

$$x_1(t) = y(t), \quad x_2(t) = \dot{y}(t), \quad \ldots, \quad x_n(t) = y^{(n-1)}(t).$$

Step 2. Write the state equations

$$\dot{x}_1(t) = x_2(t), \quad \dot{x}_2(t) = x_3(t), \quad \ldots, \quad \dot{x}_{n-1}(t) = x_n(t),$$

$$\dot{x}_n(t) = -\frac{a_0}{a_n} x_1(t) - \frac{a_1}{a_n} x_2(t) - \cdots - \frac{a_{n-1}}{a_n} x_n(t) + \frac{b_0}{a_n} u(t).$$

Step 3. Express in matrix form

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t),$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -\frac{a_0}{a_n} & -\frac{a_1}{a_n} & \cdots & -\frac{a_{n-2}}{a_n} & -\frac{a_{n-1}}{a_n} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \frac{b_0}{a_n} \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad D = [0].$$

This representation is the *companion form* (or canonical form), which is useful for analysis (stability, modes) and controller design. This works for any linear constant-coefficient ODE.

Example: 2nd-Order ODE to state-space

Mass–spring–damper ODE:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t).$$

Define states:

$$x_1 = x(t), \quad x_2 = \dot{x}(t).$$

State equations:

$$\dot{x}_1 = x_2, \qquad \dot{x}_2 = -\frac{k}{m}x_1 - \frac{c}{m}x_2 + \frac{1}{m}F(t).$$

Matrix form:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 \end{bmatrix} x.$$

Example: 3rd-Order ODE to state-space

Consider:

$$y^{(3)} + a_2 y^{(2)} + a_1 y^{(1)} + a_0 y = b_0 u(t).$$

Define states:

$$x_1 = y, \quad x_2 = \dot{y}, \quad x_3 = \ddot{y}.$$

State equations:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \dot{x}_3 = -a_0 x_1 - a_1 x_2 - a_2 x_3 + b_0 u.$$

Matrix form:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ b_0 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x.$$

Linearity in ODEs

Definition

A system is called *linear* if its equations are linear in the unknown variables and their derivatives. This means the **superposition principle** holds.

High-order Linear ODEs

An nth-order ODE of the form

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y(t) = b_0 u(t) + b_1 \frac{du}{dt} + \cdots + b_m \frac{d^m u}{dt^m}$$

is linear because

- $y(t)$ and its derivatives appear only to the first power.
- No products like y^2, $y\dot{y}$, or $\sin(y)$.
- Coefficients $\{a_i, b_i\}$ are constants (or functions of t, but not of y).

Linearity in State-Space Models

State-Space Models

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t) + Du(t)$$

This is linear because both equations are linear functions of $x(t)$ and $u(t)$. There are no products (e.g., $x_1 x_2$) or nonlinear functions (e.g., $\sin(x)$).

Superposition Principle

If $(x_1(t), y_1(t))$ is the solution for input $u_1(t)$, and $(x_2(t), y_2(t))$ is the solution for input $u_2(t)$, then for any scalars α, β:

$$u(t) = \alpha u_1(t) + \beta u_2(t)$$

produces

$$x(t) = \alpha x_1(t) + \beta x_2(t), \qquad y(t) = \alpha y_1(t) + \beta y_2(t).$$