Linear Systems on Graphs

Controllability of Structured Networks

Chapter 1

Introduction

Physical Systems and Models · From One System to Networks · Book Outline

Department of Control & Instrumentation Engineering
Pukyong National University, Busan, Korea
pnj@pknu.ac.kr

Nam-Jin Park

Chapter 1 Contents

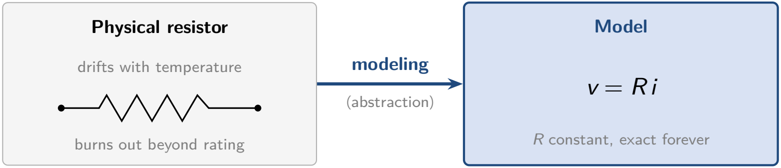
Part I: every number known
Part II: numbers gone, pattern kept
Part III: where do the inputs go?

Chapter 1: what survives when the numbers go?

From Physical System to Model

CONCEPT Modeling: Physical System → Model

- Object of analysis: **mathematical model**, not the real system
- **Resistor**: drifts with temperature, burns out beyond rating → **constant** R
- **Robot link**: flexes under load → **perfectly rigid**
- **How far can a later result be trusted?** → **only as far as the model**



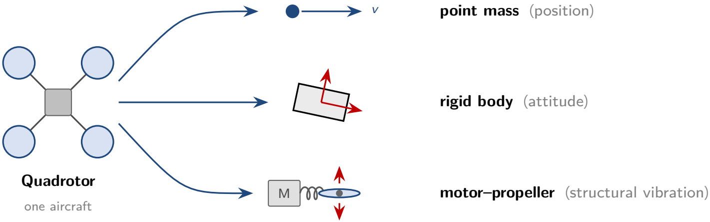
Convention of This Course

- "System" from here on = **a model of a physical system**

One Physical System, Three Models

EXAMPLE Quadrotor

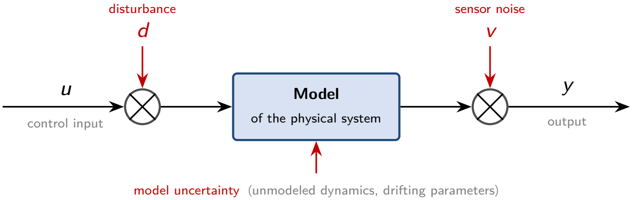
- To a **path planner**: a **point mass**
- To an **attitude controller**: a **rigid body**
- To a **vibration analyst**: **coupled motor–propeller dynamics**



One system, **more than one model**, no conflict. **Which model is right?** $\rightarrow$ the one matching the **control input** and the **output of interest**.

The Modeling Gap: Disturbance and Uncertainty

- **Inputs**: **control** u (chosen) + **disturbance** d (not chosen, not measured)
- Output read through a sensor → measurement carries **noise** v
- Nonlinear system → **linearization** about an **operating point** → a linear model, trustworthy only near that point



[The modeling gap] No model is exact. **How close must it be?** → small error over the **operating range of interest** → modeling assumed done, the model is the starting point.

Four Stages of Analytical Study: Where This Course Lives

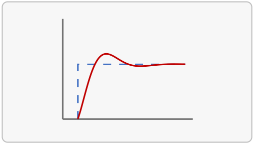
Modeling → Description → **Analysis** → Design

- **Modeling**: physical system → model
- **Description**: model → mathematical equations
- **Analysis**: predict and evaluate behavior
- **Design**: modify to meet specs

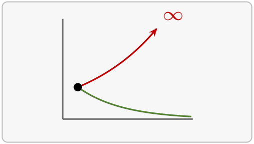
▶ Two Kinds of Analysis

- **Whether** first: is it stable? can the input steer every state? does the output reveal the state? → **yes or no**
- **How much** next: how stable, how fast, how large is the response? → **a number or a curve**

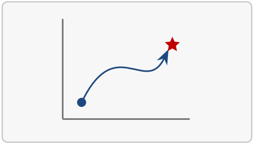
state = the variables carrying the system's memory → state now + input from now on fixes the whole future.



what response?



stable?

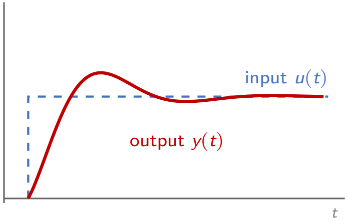


controllable?

Response and Stability: Two Analysis Questions

CONCEPT Response (How much)

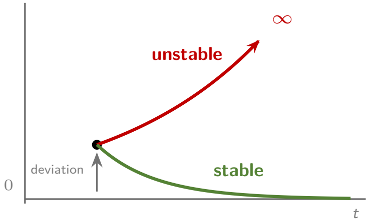
- **Given**: model + input $u(t)$
- **Ask**: which output $y(t)$ follows? $\rightarrow$ **a curve**



model + input
$\Rightarrow$ one definite curve

CONCEPT Stability (Whether)

- **Given**: model + an *initial deviation* from **equilibrium** (the state it stays at if undisturbed)
- **Ask**: **converge** back, or **diverge**? $\rightarrow$ **yes or no**



same initial deviation
$\Rightarrow$ converge, or diverge

How much? $\rightarrow$ **a curve**. *Whether*? $\rightarrow$ **yes or no**. Design *insight*: the second.

Interlude: When the Answer Is “Not Stable”

Tacoma Narrows Bridge, 1940

www.youtube.com/watch?v=mXTSnZgrfxM

Practical Engineering, *Why the Tacoma Narrows Bridge Collapsed*

- Every computed load carried $\rightarrow$ the **How much** answers not wrong
- Small deviation grew on its own until collapse $\rightarrow$ the **Whether** answer: *not stable*
- Mechanism = **flutter**, not forced resonance: airflow feeds energy back into the motion $\rightarrow$ wind + deck = **feedback loop** $\rightarrow$ unstable

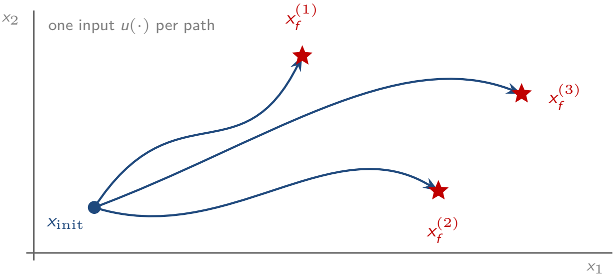
  Which of the two analyses would have caught this one?

Controllability: Can the Input Steer the State Anywhere?

CONCEPT Controllability (Whether)

- **Given**: model, start x_{init}, *any* target x_f
- **Ask**: input $u(\cdot)$ driving $x_{\text{init}} \to x_f$ in **finite time**, for *every* target? $\to$ **yes or no**

State = **vector** $x = (x_1, x_2)^\top$, $n = 2$ **components** $\to$ each point of the plane is one state.



every target reachable in finite time $\Rightarrow$ controllable

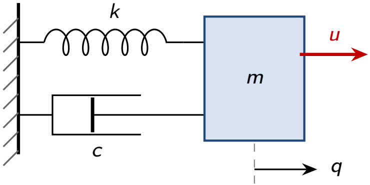
One property throughout: **controllability**. How much of the model must be known to answer it? $\to$ **progressively weaker knowledge**, chapter by chapter.

Chapter 1 Contents

The Classical World: One System, Every Number Known

- Classical control = **one system, parameters known** (bench · data · datasheet)
- **Part I**: that setting exactly → **every number in the model trusted**
- Examples: mass–spring–damper, DC motor → m, c, k, R, L all known

EXAMPLE Mass–spring–damper



q measured from equilibrium

- Newton's second law: spring $k\,q$ · damper $c\,\dot{q}$
- **One second-order** equation in q

$$m\,\ddot{q} + c\,\dot{q} + k\,q = u$$

$\ddot{q}$ acceleration · $\dot{q}$ velocity · q position

Measure m, c, k once → **every coefficient fixed**. Next: one second-order equation → **first-order** system, the form Part I uses.

Second-Order Equation $\rightarrow$ $\dot{x} = Ax + Bu$

▶ Step 1: Choose the state

$$x_1 = q \ \text{(position)}, \qquad x_2 = \dot{q} \ \text{(velocity)}, \qquad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

▶ Step 2: Write one first-order equation for each component

$$\dot{x}_1 = \dot{q} = x_2, \qquad \dot{x}_2 = \ddot{q} = \frac{1}{m}\big(u - c\,\dot{q} - k\,q\big) = -\frac{k}{m}\,x_1 - \frac{c}{m}\,x_2 + \frac{1}{m}\,u$$

▶ Step 3: Stack the two equations

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ 1/m \end{bmatrix}}_{B} u, \qquad y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} x = q$$

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

state-space form · A the system · B the input · C the output, what the sensor reads

[The construction] Linear system $\rightarrow$ n first-order equations, one per **state**. After this slide: the pair (A, B), with C where the output matters.

Networked Systems: One Matrix, One Graph

- **Network = simple units + interconnection structure** (power grid, robot team, gene regulatory network) → wiring diagram → **which edges exist: known exactly**

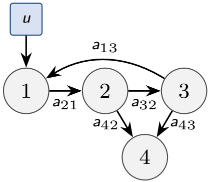
$$\underbrace{\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}0 & 0 & a_{13} & 0\\ a_{21} & 0 & 0 & 0\\ 0 & a_{32} & 0 & 0\\ 0 & a_{42} & a_{43} & 0\end{bmatrix}}_{A} \underbrace{\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}}_{x} + \underbrace{\begin{bmatrix}1\\ 0\\ 0\\ 0\end{bmatrix}}_{b} u$$

$$\dot{x}_1 = a_{13}\, x_3 + u$$
$$\dot{x}_2 = a_{21}\, x_1$$
$$\dot{x}_3 = a_{32}\, x_2$$
$$\dot{x}_4 = a_{42}\, x_2 + a_{43}\, x_3$$

$x_i \mapsto$ **node** i

$a_{ij} \neq 0 \mapsto$ **edge** $j \to i$

$b \mapsto$ **input column**
(single input)

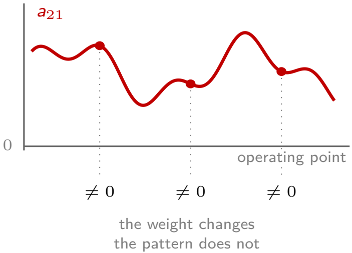


[The correspondence] Matrix and picture: **one object**. **Can the five weights be trusted?** → **next slide**.

Weights Drift: Numerical Answers Expire

CONCEPT Values Move, the Pattern Stays

- **Edge weight** a_{ij} $\rightarrow$ drifts with **operating point**, load, temperature
- **Edge existence** (does the relation exist at all?) $\rightarrow$ **fixed**
- Answer from *today's* **weights** $\rightarrow$ **expires tomorrow**. Existence $\rightarrow$ stays



Keep Only the Pattern

- Every entry: **structural zero** (no edge) or **free non-zero** parameter
- Such a matrix = **pattern matrix** $\rightarrow$ its network = **structured network**
- Weights erased $\rightarrow$ one **pattern graph** serves every **realization** (one choice of the free weights, Ch. 4)

[Premise of Part II] Numbers in A untrusted, **zero/non-zero pattern** trusted.
How can we judge controllability?

Teaser: One Number Flips the Answer

EXAMPLE **Same pattern, opposite answers (formally in week 3)**

- **Identical pattern**: same nodes, same edges, $b = (1,1)^\top$ $\rightarrow$ only a_{21} differs

▶ The Test: Kalman Rank Condition

$$\mathcal{C} = \begin{bmatrix} b & Ab & \cdots & A^{n-1}b \end{bmatrix}, \qquad \text{controllable} \iff \operatorname{rank}\mathcal{C} = n$$

$\mathcal{C}$ the **controllability matrix** · rank the number of **independent columns**

here $n = 2$ $\rightarrow$ $\mathcal{C} = \begin{bmatrix} b & Ab \end{bmatrix}$

$$A_\circ = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}, \quad A_\circ b = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \qquad\qquad A_\bullet = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix}, \quad A_\bullet b = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

$$\mathcal{C} = \begin{bmatrix} 1 & 3 \\ 1 & 3 \end{bmatrix}, \quad \det\mathcal{C} = 0 \qquad\qquad \mathcal{C} = \begin{bmatrix} 1 & 3 \\ 1 & 5 \end{bmatrix}, \quad \det\mathcal{C} = 2$$

$\operatorname{rank}\mathcal{C} = 1 < 2$ $\rightarrow$ **uncontrollable** $\qquad$ $\operatorname{rank}\mathcal{C} = 2$ $\rightarrow$ **controllable**

Network $\rightarrow$ **weights unknown**, **zeros certain**. **What can the pattern alone guarantee?**
$\Rightarrow$ Part II.

Chapter 1 Contents

Three Parts, One Chain of Questions

Part I Linear Systems Theory (Ch. 1–3): state-space models and **their graphs** · controllability & observability · Kalman, Gramian, PBH tests (Ch. 3)

↓ what can the pattern guarantee?

Part II Analysis (Ch. 4–8): **pattern graph** (4) → structural vs strong structural (5) → subspaces (6) → graph criteria (7) → fixed nodes (8)

↓ where should the inputs go?

Part III Synthesis (Ch. 9–10): **minimum inputs** → **optimal placement**

- Each chapter opens on **the question the previous one left**
- **Where does the network in your own research sit on this chain?**

Two Axes Running Through the Book

- **Axis 1: SC vs SSC**
  - **Structural Controllability (SC)**: controllable for **almost all realizations**
  - **Strong Structural Controllability (SSC)**: controllable for **all realizations**

$\mathcal{C}_F = [\, b \;\; A_F b \;\; \cdots \,]$ at one **realization** F. One input → square → $\det \mathcal{C}_F$ a polynomial in the free **edge weights**. SC → vanishes on a **measure-zero** set. SSC → never vanishes.

chain $u \to 1 \to 2 \to 3$

$$A_F = \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ 0 & a_{32} & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\det \mathcal{C}_F = a_{21}^2\, a_{32}$$

non-zero **product** → never zero → **SSC**

flip two nodes, $b = (1,1)^\top$ (the teaser)

$$A_F = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\det \mathcal{C}_F = a_{21} + a_{22} - a_{11} - a_{12}$$

sum → can be zero → **SC**, not SSC

- **Axis 2: Algebra vs Graph**
  - Every condition stated in **both forms**: determinant · rank ↔ stems · matchings · zero forcing (the graph words, each defined in Part II)

Prerequisites and Next Week

- **Prerequisites (all reviewed in week 2)**
  - Undergraduate control · linear algebra (eigenvalues, rank) · MATLAB
  - Graph theory: **not assumed** $\rightarrow$ introduced within the course
- **Next Week (Ch. 2: Foundations)**
  - State-space models + the **graph of a linear system** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - Linear-algebra background · solutions and stability · Laplacian dynamics and consensus (one-page roadmap: Preface)

Bring one network you care about: which of its edges do you actually know?

> **[Line of the day]** "Every linear system is already a graph.
> Part II merely **erases the numbers** on the edges."