

AA203

Optimal and Learning-based Control

Course overview; control, stability, performance metrics



Stanford
University



Course mechanics

Teaching team:

- Instructor: Ed Schmerling (OH: M 1pm-2pm; Project OH: W 4pm-5pm)
- CAs: Spencer M. Richards and Devansh Jalota (OH: TBD)

Logistics:

- Lecture slides, homework assignments: <http://asl.stanford.edu/aa203/>
- Lecture recordings, announcements: <https://canvas.stanford.edu/courses/151934>
- Discussion forum: <https://edstem.org/us/courses/21368>
- Homework submission: <https://www.gradescope.com/courses/380844>
- For urgent questions: aa203-spr2122-staff@lists.stanford.edu

Course requirements

- Homework: there will be a total of four graded problem sets
 - Mixture of theory and implementation (Python)
- Final project: details on the course website
 - Open-ended, groups of (up to) 3 people
- Grading:
 - Homework: 60% (15% per HW)
 - Final project: 40% (5% proposal, 10% midterm report, 25% final report)
 - Ed Discussion: bonus up to 5%, 0.5% per endorsed post
- Late day policy: 6 total, maximum of 3 on any given assignment

Course material

- Course notes: an evolving set of partial course notes is available at <https://github.com/StanfordASL/AA203-Notes>
- Recitations: Friday lecture sessions (F 9:45-11:15AM, weeks 1-4) led by the CAs covering relevant tools (computational and mathematical)
- Textbooks that may be valuable for context or further reference are listed in the syllabus

Prerequisites

- Familiarity with a standard undergraduate engineering mathematics curriculum (e.g., CME100-106; vector calculus, ordinary differential equations, introductory probability theory)
- **Strong** familiarity with linear algebra (e.g., EE263 or CME200)
- Nice-to-have: a course in optimization (e.g., EE364A, CME307, CS269O, AA222)
- To get the most out of this class, at least one of:
 - A course in machine learning (e.g., CS229, CS230, CS231N)
or
 - A course in control (e.g., ENGR105, ENGR205, AA212)

Homework 0 (ungraded) is out now to help you gauge your preparedness.

Caveats

- Arguably, this class aims for “breadth over depth”
 - Past students have found self-study of the details necessary
- The subject matter is a moving target
 - Past students have lamented bugs in new HW problems
- This class is quite challenging/demanding
 - Past students have noted that project progress is difficult to pace with HWs
- Projects focused on learning-based control may require some self-study before the relevant lectures (talk to the teaching staff for pointers)

Today's Outline

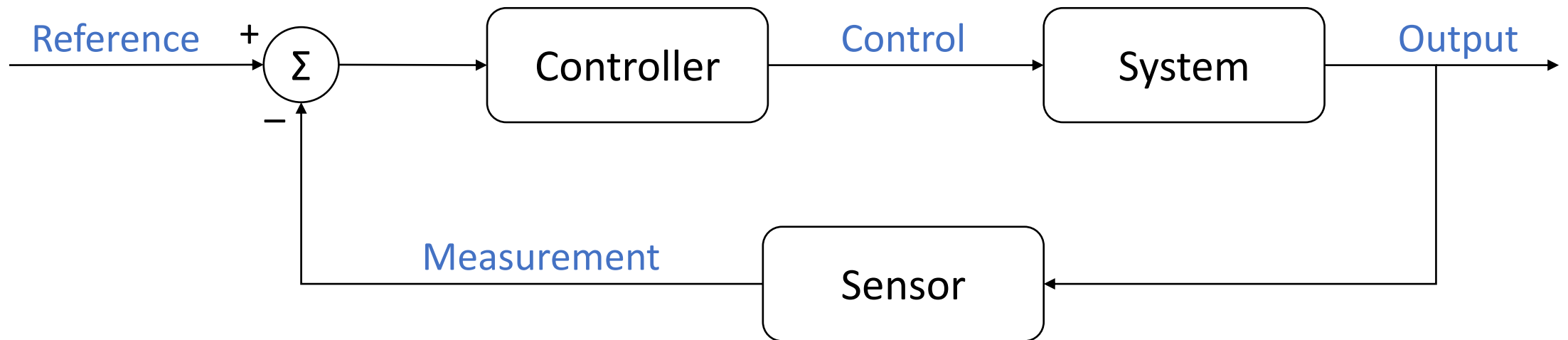
1. Context and course goals
2. State-space models
3. Problem formulation for optimal control

Today's Outline

1. Context and course goals
2. State-space models
3. Problem formulation for optimal control

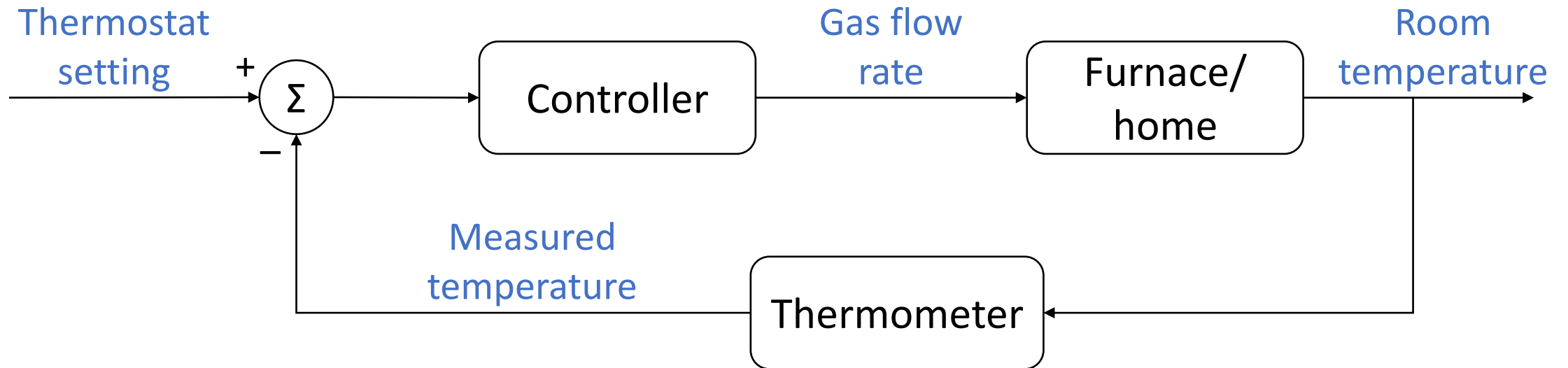
Feedback control

- Tracking a reference signal



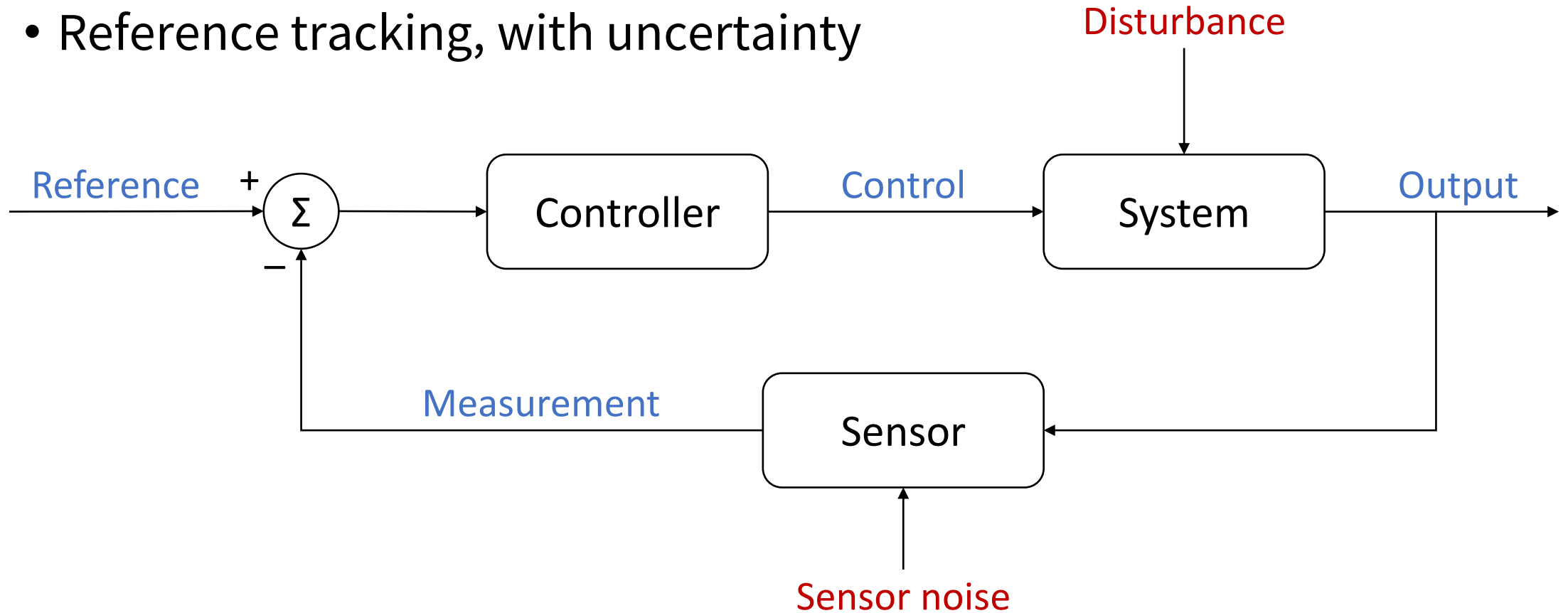
Feedback control

- Tracking a reference signal



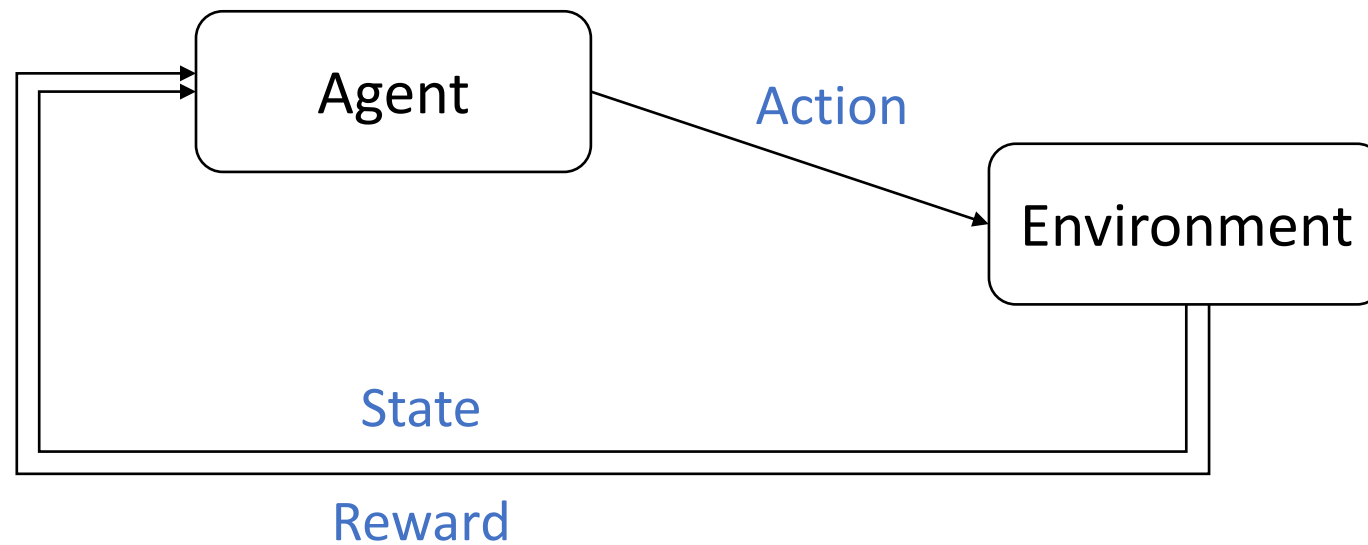
Feedback control

- Reference tracking, with uncertainty



Reinforcement learning

- A brief aside...



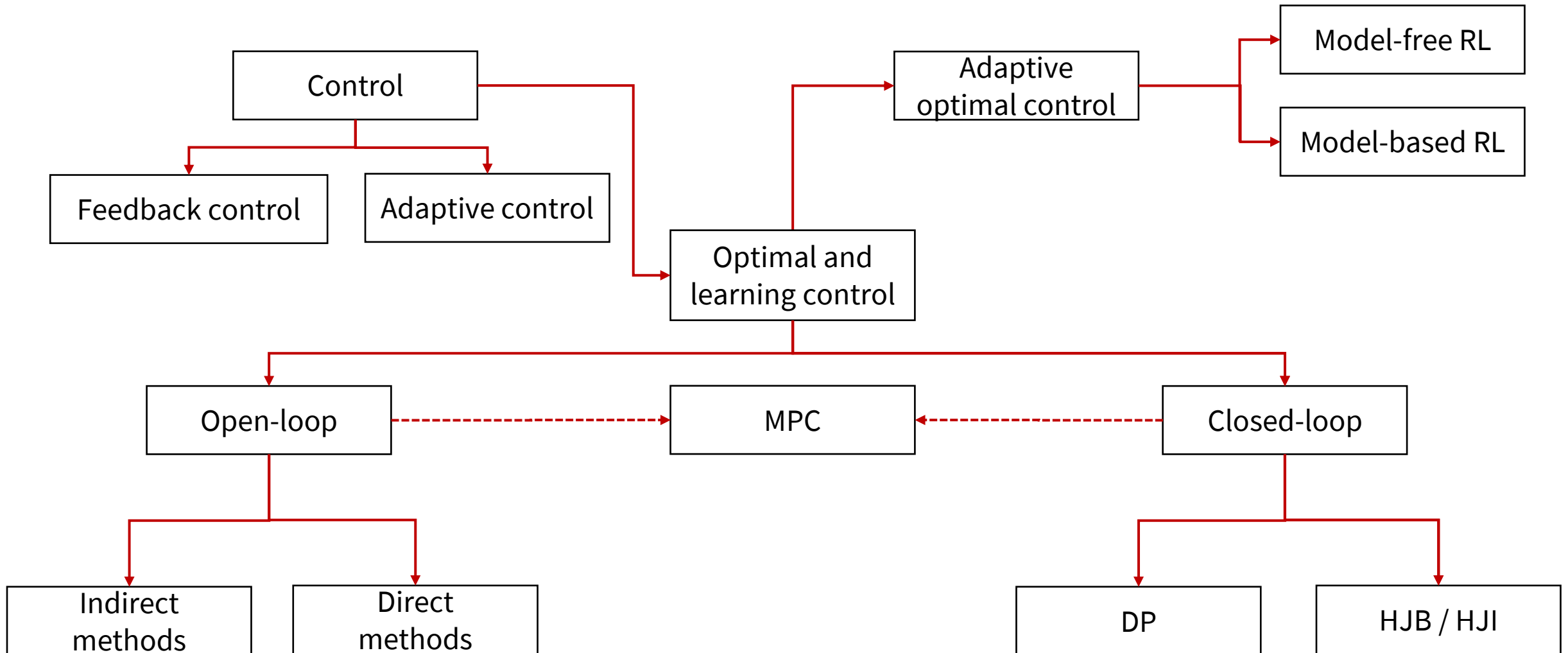
Feedback control desiderata

- Stability: multiple notions; loosely system output is “under control”
- Tracking: the output should track the reference “as closely as possible”
- Disturbance rejection: the output should be “as insensitive as possible” to disturbances/noise
- Robustness: controller should still perform well up to “some degree of” model misspecification

What's missing?

- Performance: mathematical quantification of the above desiderata, and providing a control that best realizes the tradeoffs between them
- Planning: providing an appropriate reference trajectory for the controller to track (particularly nontrivial, e.g., when controlling mobile robots)
- Learning: a controller that adapts to an initially unknown, or possibly time-varying system

Course overview



Course goals

To learn the *theoretical* and *implementation* aspects of main techniques in **optimal and learning-based control**

Course goals

To learn the *theoretical* and *implementation* aspects of main techniques in **optimal and learning-based control**

To provide a *unified framework and context* for understanding and relating these techniques to each other

Today's Outline

1. Context and course goals
2. State-space models
3. Problem formulation for optimal control

Mathematical model

$$\dot{x}_1(t) = f_1(x_1(t), x_2(t), \dots, x_n(t), u_1(t), u_2(t), \dots, u_m(t), t)$$

$$\dot{x}_2(t) = f_2(x_1(t), x_2(t), \dots, x_n(t), u_1(t), u_2(t), \dots, u_m(t), t)$$

\vdots \vdots

$$\dot{x}_n(t) = f_n(x_1(t), x_2(t), \dots, x_n(t), u_1(t), u_2(t), \dots, u_m(t), t)$$

Where

- $x_1(t), x_2(t), \dots, x_n(t)$ are the state variables
- $u_1(t), u_2(t), \dots, u_m(t)$ are the control inputs

Mathematical model

In compact form

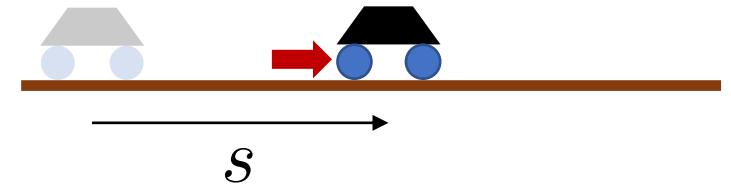
$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$$

- a history of control input values during the interval $[t_0, t_f]$ is called a *control history*
- a history of state values during the interval $[t_0, t_f]$ is called a *state trajectory*

Illustrative example

- Double integrator: point mass under controlled acceleration

$$\ddot{s}(t) = a(t)$$

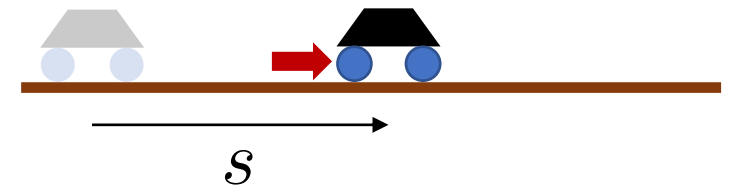


Illustrative example

- Double integrator: point mass under controlled acceleration

$$\ddot{s}(t) = a(t)$$

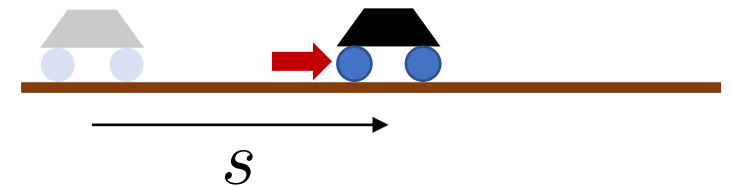
$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ a \end{bmatrix}$$



Example system

- Double integrator: point mass under controlled acceleration

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [a]$$

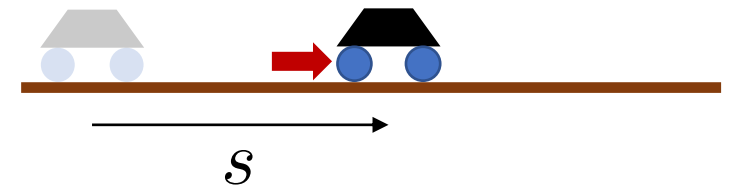


Example system

- Double integrator: point mass under controlled acceleration

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [a]$$

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



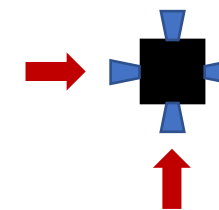
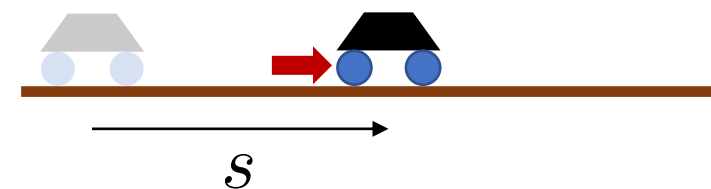
Example system

- Double integrator: point mass under controlled acceleration

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [a]$$

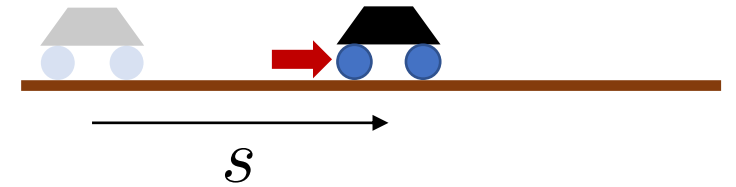
$$\dot{\mathbf{x}}(t) = A \mathbf{x}(t) + B \mathbf{u}(t)$$

$$\begin{bmatrix} \dot{\mathbf{s}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{v} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} [\mathbf{a}]$$



Example controller

Let's drive from $[5, 0]^T$ to $[0, 0]^T$.

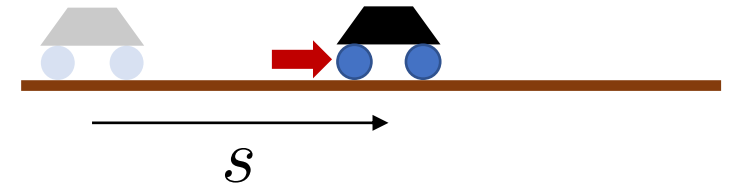


Example controller

Let's drive from $[5, 0]^T$ to $[0, 0]^T$.

Proposal: use a linear feedback control law.

$$a = -k_p s - k_d v$$



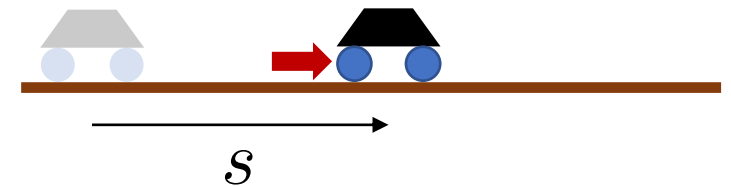
Example controller

Let's drive from $[5, 0]^T$ to $[0, 0]^T$.

Proposal: use a linear feedback control law.

$$a = -k_p s - k_d v$$

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_p & k_d \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix}$$



Example controller

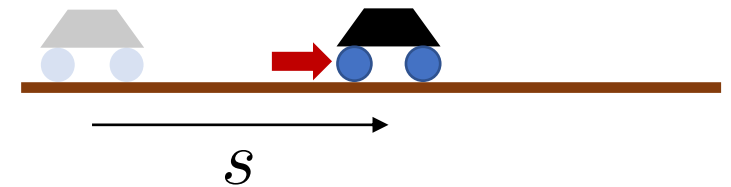
Let's drive from $[5, 0]^T$ to $[0, 0]^T$.

Proposal: use a linear feedback control law.

$$a = -k_p s - k_d v$$

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_p & k_d \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix}$$

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix}$$



Example controller

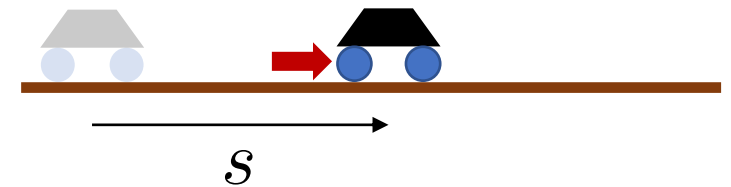
Let's drive from $[5, 0]^T$ to $[0, 0]^T$.

Proposal: use a linear feedback control law.

$$a = -k_p s - k_d v$$

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_p & k_d \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix}$$

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} \quad \left(\dot{\mathbf{x}}(t) = (A - BK)\mathbf{x}(t) \right)$$



Analyzing stability

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix}$$

Analyzing stability

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix}$$

$$\begin{bmatrix} s(t) \\ v(t) \end{bmatrix} = \exp \left(\begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} t \right) \begin{bmatrix} s(0) \\ v(0) \end{bmatrix}$$

Analyzing stability

$$\begin{bmatrix} \dot{s} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix}$$

$$\begin{bmatrix} s(t) \\ v(t) \end{bmatrix} = \exp\left(\begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} t\right) \begin{bmatrix} s(0) \\ v(0) \end{bmatrix}$$

$$\begin{bmatrix} s(t) \\ v(t) \end{bmatrix} = V^{-1} \begin{bmatrix} e^{\lambda_+ t} & 0 \\ 0 & e^{\lambda_- t} \end{bmatrix} V \begin{bmatrix} s(0) \\ v(0) \end{bmatrix}$$

$$\text{where } \lambda_{\pm} = \left(-k_d \pm \sqrt{k_d^2 - 4k_p}\right) / 2$$

or

$$\begin{bmatrix} s(t) \\ v(t) \end{bmatrix} = e^{\lambda t} V^{-1} \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} V \begin{bmatrix} s(0) \\ v(0) \end{bmatrix}$$

$$\text{where } \lambda = -k_d/2, \text{ if } k_d^2 - 4k_p = 0$$

Analyzing stability

$$\begin{bmatrix} s(t) \\ v(t) \end{bmatrix} = V^{-1} \begin{bmatrix} e^{\lambda_+ t} & 0 \\ 0 & e^{\lambda_- t} \end{bmatrix} V \begin{bmatrix} s(0) \\ v(0) \end{bmatrix}$$

or

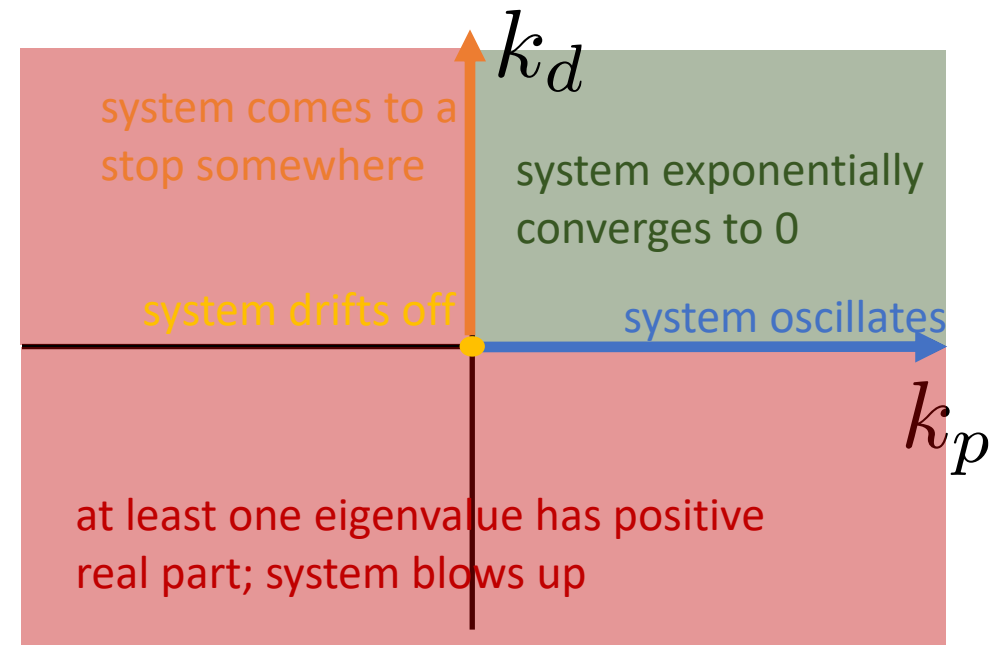
$$\begin{bmatrix} s(t) \\ v(t) \end{bmatrix} = e^{\lambda t} V^{-1} \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} V \begin{bmatrix} s(0) \\ v(0) \end{bmatrix}$$

where $\lambda_{\pm} = \left(-k_d \pm \sqrt{k_d^2 - 4k_p} \right) / 2$

where $\lambda = -k_d/2$, if $k_d^2 - 4k_p = 0$

$\text{Re}(\lambda) \rightarrow$ exponential growth (> 0),
exponential decay (< 0),
or constant ($=0$)

$\text{Im}(\lambda) \rightarrow$ sinusoidal oscillation



Mathematical definitions of stability

Many notions:

- Asymptotic stability
 - Global: all trajectories converge to the equilibrium
 - Local: all trajectories starting near the equilibrium converge to the equilibrium
- Exponential stability
 - Same as asymptotic stability, but with exponential rate
- Marginal stability
- Bounded-input, bounded-output stability
- Lyapunov stability

Quantifying performance

$$\begin{aligned} \min & \int_0^{t_f} \|\mathbf{x}(t)\|_2^2 + \|\mathbf{u}(t)\|_2^2 dt \\ \text{s.t.} & \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ & \mathbf{x}(0) = \mathbf{x}_0 \end{aligned}$$

Quantifying performance

$$\begin{aligned} \min \int_0^{t_f} & \|\mathbf{x}(t)\|_2^2 + \|\mathbf{u}(t)\|_2^2 dt \\ \text{s.t. } \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{x}(0) &= \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \end{aligned}$$

Quantifying performance

$$\min \int_0^{t_f} \mathbf{x}(t)^T Q \mathbf{x}(t) + \mathbf{u}(t)^T R \mathbf{u}(t) dt$$

$$\text{s.t. } \dot{\mathbf{x}}(t) = A \mathbf{x}(t) + B \mathbf{u}(t)$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f$$

Quantifying performance

$$\begin{aligned} \min \quad & \int_0^{t_f} \mathbf{x}(t)^T Q \mathbf{x}(t) + \|\mathbf{u}(t)\|_1 dt \\ \text{s.t.} \quad & \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ & \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \end{aligned}$$

Today's Outline

1. Context and course goals
2. State-space models
3. Problem formulation for optimal control

Problem formulation

- Mathematical description of the system to be controlled
- Statement of the constraints
- Specification of a performance criterion

Performance measure

$$J = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t) dt$$

- h (terminal cost) and g (stagewise/running cost) are scalar functions
- t_f may be specified or free

Constraints

- initial and final conditions (boundary conditions)

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f$$

- constraints on state trajectories

$$\underline{X} \leq \mathbf{x}(t) \leq \overline{X}$$

- control authority

$$\underline{U} \leq \mathbf{u}(t) \leq \overline{U}$$

- and many more...

Constraints

- A control history which satisfies the control constraints during the entire time interval $[t_0, t_f]$ is called an **admissible control**
- A state trajectory which satisfies the state variable constraints during the entire time interval $[t_0, t_f]$ is called an **admissible trajectory**

Optimal control problem

Find an *admissible control* \mathbf{u}^* which causes the system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$$

to follow an *admissible trajectory* \mathbf{x}^* that minimizes the performance measure

$$J = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t) dt$$

Optimal control problem

Comments:

- minimizer $(\mathbf{x}^*, \mathbf{u}^*)$ called optimal trajectory-control pair
- existence: in general, not guaranteed
- uniqueness: optimal control may not be unique
- minimality: we are seeking a global minimum
- for maximization, we rewrite the problem as $\min_{\mathbf{u}} -J$

Form of optimal control

1. if $\mathbf{u}^* = \pi(\mathbf{x}(t), t)$, then π is called optimal control law or optimal policy (*closed-loop*)
 - important example: $\pi(\mathbf{x}(t), t) = F \mathbf{x}(t)$
2. if $\mathbf{u}^* = e(\mathbf{x}(t_0), t)$, then the optimal control is *open-loop*
 - optimal *only* for a particular initial state value

Discrete-time formulation

- **System:** $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k)$, $k = 0, \dots, N - 1$
- **Control constraints:** $\mathbf{u}_k \in U$
- **Cost:**

$$J(\mathbf{x}_0; \mathbf{u}_0, \dots, \mathbf{u}_{N-1}) = h_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} g_k(\mathbf{x}_k, \mathbf{u}_k, k)$$

- **Decision-making problem:**

$$J^*(\mathbf{x}_0) = \min_{\mathbf{u}_k \in U, k=0, \dots, N-1} J(\mathbf{x}_0; \mathbf{u}_0, \dots, \mathbf{u}_{N-1})$$

Discrete-time formulation

- **System:** $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k)$, $k = 0, \dots, N - 1$
- **Control constraints:** $\mathbf{u}_k \in U$
- **Cost:**

$$J(\mathbf{x}_0; \mathbf{u}_0, \dots, \mathbf{u}_{N-1}) = h_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} g_k(\mathbf{x}_k, \mathbf{u}_k, k)$$

- **Decision-making problem:**

$$J^*(\mathbf{x}_0) = \min_{\mathbf{u}_k \in U, k=0, \dots, N-1} J(\mathbf{x}_0; \mathbf{u}_0, \dots, \mathbf{u}_{N-1})$$

Extension to stochastic setting will be covered later in the course

Next class

Introduction to learning;
System identification and adaptive control