AA 203
Optimal and Learning-Based Control
Course overview; Feedback, stability, and optimal control problems

Spencer M. Richards

Autonomous Systems Laboratory, Stanford University

April 3, 2023
(last updated May 3, 2023)

Teaching team

			
Spencer M. Richards (Instructor)	Daniele Gammelli (Instructor)	Thomas Lew (CA)	Devansh Jalota (CA)

Important links

- Lecture slides and homework assignments:
  https://asl.stanford.edu/aa203
- Lecture recordings:
  https://canvas.stanford.edu/courses/171491
- Announcements and discussion forum:
  https://edstem.org/us/courses/38294
- Coursework submission:
  https://www.gradescope.com/courses/525712
- For urgent questions:
  aa203-spr2223-staff@lists.stanford.edu

Grading

Homework (60%)
- 4 homeworks, each problem weighted equally across all homeworks.
- Covers a mixture of theory and programming.
- Generally due every 2 weeks.

Project (40%)
- 5% proposal, 10% midterm report, 25% final report and video presentation.
- Open-ended in groups of up to 3 people.

Discussion ($\leq$ 5% bonus)
- 0.5% per endorsed Edstem post, up to 5%.

Late days
- 6 total, up to 3 on a single assignment.
- *Not applicable* to the final report and video presentation (due on the last day of class).

Course material

In order of importance:

- **Lecture slides** Should be posted on the class website before each lecture.
- **Recitations** Friday lecture sessions (Weeks 1–4) led by the CAs covering supplementary tools (mathematical and computational).
- **Course notes** Evolving, somewhat outdated partial notes available at: `https://github.com/StanfordASL/AA203-Notes`
- **Textbooks** Suggested ad hoc during lecture and discussions (not required).

Prerequisites

- Standard undergraduate engineering mathematics knowledge (i.e., vector calculus, ordinary differential equations (ODEs), probability theory).
- *Strong* familiarity with linear algebra (e.g., EE263, CME200).
- Some knowledge of optimization is nice to have (e.g., EE364A, CME307, CS269O, AA222).
- To get the most out of this class, it is recommended to have taken at least one course in:
  - control (e.g., ENGR105, ENGR205, AA212)
  - machine learning (e.g., CS229, CS230, CS231N)
- *Homework 0 (ungraded)* is out now to help you gauge your preparedness.

Caveats

- Arguably, this class aims for breadth over depth. Some past students have needed to self-study some of the details.
- The course content is subject to feedback. Homework problems covering state-of-the-art topics sometimes suffer from bugs.
- This class is quite challenging. Some past students have had trouble managing both homeworks and project deliverables.
- Projects focused on *learning-based* control may require self-study of material before the relevant lectures.

Agenda

1. Context and course goals

2. Stability and Lyapunov functions

3. Optimal control problems

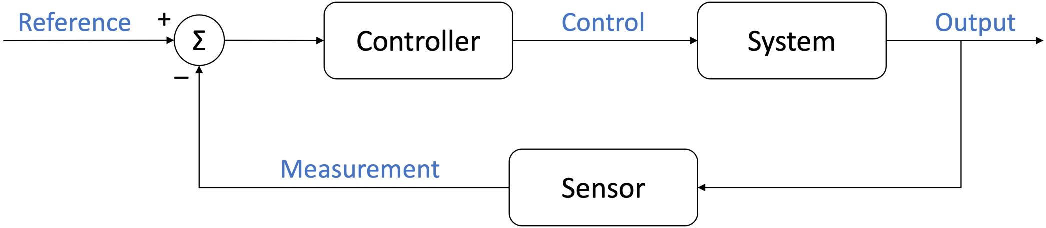
Agenda

1. Context and course goals

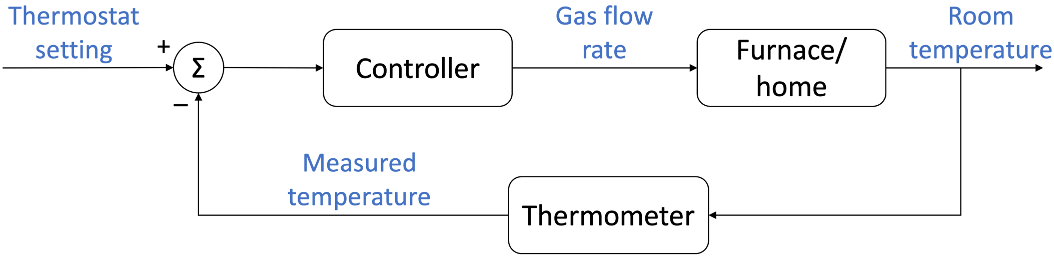
2. Stability and Lyapunov functions

3. Optimal control problems

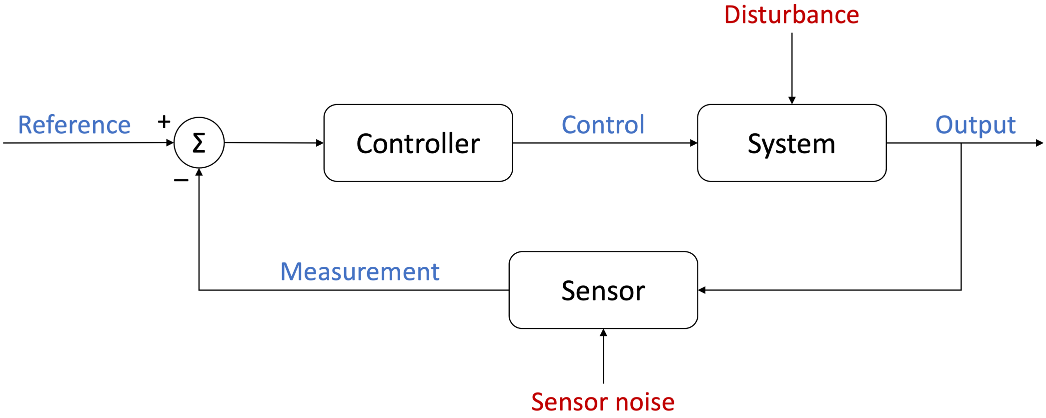
Feedback control



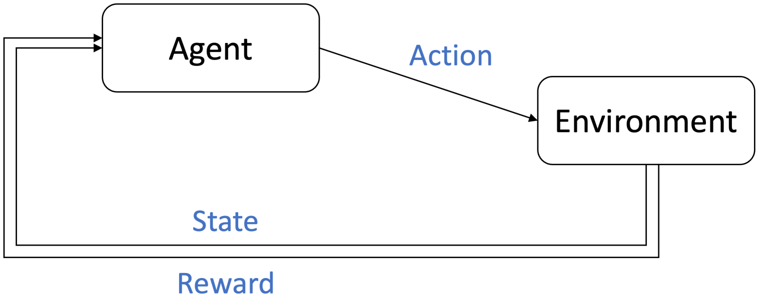
Feedback control example



Feedback control with complications



Feedback control in reinforcement learning



Dynamical systems

	Continuous-time:	Discrete-time:
Time	$t \in \mathbb{R}$	$t \in \mathbb{N}$
State	$x(t) \in \mathbb{R}^n$	$x_t \in \mathbb{R}^n$
Control input	$u(t) \in \mathbb{R}^m$	$u_t \in \mathbb{R}^m$
Dynamics	$\dot{x}(t) = f(t, x(t), u(t))$	$x_{t+1} = f(t, x_t, u_t)$
Trajectories	$x : t \mapsto x(t)$ $u : t \mapsto u(t)$	$x : t \mapsto x_t$ $u : t \mapsto u_t$

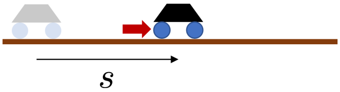
We assume f is sufficiently "well-behaved" such that, given a piecewise-continuous input u, there exists a unique solution x for each initial condition.

In roughly the second-half of the course, the dynamics may be *unknown*, and so will have to *learn* how to control our system based on data.

Example: Double-integrator control

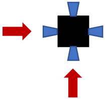
Point-mass with acceleration control in 1-D:

$$\begin{pmatrix}\dot{s}\\ \dot{v}\end{pmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{pmatrix}s\\ v\end{pmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u$$



More generally, in multiple dimensions we have:

$$\begin{pmatrix}\dot{s}\\ \dot{v}\end{pmatrix} = \begin{bmatrix}0 & I\\ 0 & 0\end{bmatrix}\begin{pmatrix}s\\ v\end{pmatrix} + \begin{bmatrix}0\\ I\end{bmatrix}u$$



Objective Drive to a standstill at the origin, i.e., $(0, 0)$.

Proposal Proportional-derivative (PD) feedback:

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

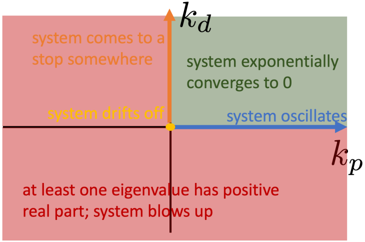
Example: Double-integrator stability

Is the closed-loop system stable?

$$\begin{pmatrix} \dot{s} \\ \dot{v} \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} \begin{pmatrix} s \\ v \end{pmatrix} \implies \begin{pmatrix} s(t) \\ v(t) \end{pmatrix} = \underbrace{\exp\left(\begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} t \right)}_{=:\Phi(t)} \begin{pmatrix} s(0) \\ v(0) \end{pmatrix}$$

where $\Phi(t) = V \exp(tJ) V^{-1}$ with eigenvalues $\lambda_\pm = -\frac{k_d}{2} \pm \frac{1}{2}\sqrt{k_d^2 - 4k_p}$ and

$$\exp(tJ) = \begin{cases} \begin{bmatrix} e^{\lambda_+ t} & 0 \\ 0 & e^{\lambda_- t} \end{bmatrix}, & k_d^2 > 4k_p \\ \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} e^{-\frac{k_d}{2} t}, & k_d^2 = 4k_p \\ \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix} e^{-\frac{k_d}{2} t}, & k_d^2 < 4k_p \end{cases}$$



Feedback control desiderata

Traditional feedback control balances the following desiderata.

Stability	The system output does not diverge or "blow up".
Tracking	The system output converges to a desired reference.
Disturbance rejection	The system is insensitive to disturbances and noise.
Robustness	The controller performs well despite some model misspecification.

Optimal and learning-based control desiderata

This course also incorporates and focuses on the following objectives.

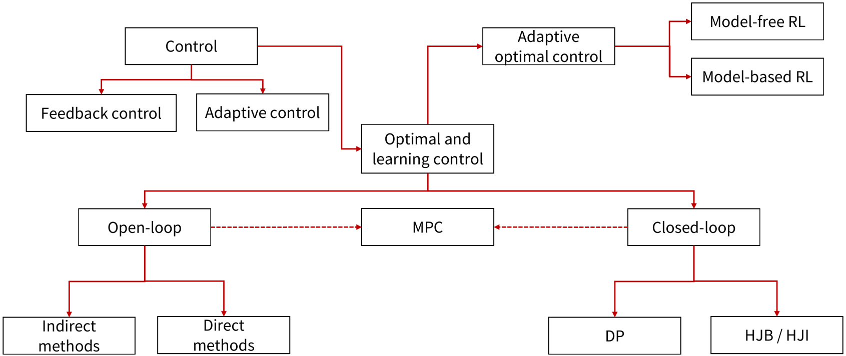
Performance The controller achieves an optimal trade-off between various metrics.

Constraints The controller does not cause the system to violate safety restrictions or inherent (e.g., physical) limitations.

Planning An appropriate reference trajectory is computed and given to the controller for tracking.

Learning The controller can adapt to an unknown or time-varying system.

Course overview and goals



- To learn the *theory* and *practice* of fundamental techniques in optimal and learning-based control.
- To gain a *holistic understanding* of how such techniques are used across fields.

Agenda

Mathematical definitions of stability

Consider $\dot{x} = f(x)$ (or $\dot{x} = f(x, \pi(x))$) and an *equilibrium* $\bar{x} \in \mathbb{R}^n$ (i.e., $f(\bar{x}) = 0$).

Marginal/Lyapunov $\forall \varepsilon > 0, \exists \delta > 0 : \|x(0) - \bar{x}\| < \delta \implies \|x(t) - \bar{x}\| < \varepsilon, \ \forall t \geq 0$
"Trajectories that start close to the equilibrium remain close to the equilibrium."

Asymptotic (local) $\exists \delta > 0 : \|x(0) - \bar{x}\| < \delta \implies \lim_{t \to \infty} \|x(t) - \bar{x}\| = 0$
"Trajectories that start near the equilibrium converge to it."

Exponential (local) $\exists \delta, c, \alpha > 0 : \|x(0) - \bar{x}\| < \delta \implies \|x(t) - \bar{x}\| \leq c e^{-\alpha t} \|x(0) - \bar{x}\|$
"Trajectories that start near the equilibrium converge to it exponentially fast."

Take $\delta \to \infty$ to get "global" definitions. For linear time-invariant (LTI) systems, "asymptotic = exponential" and "local = global" always.

Lyapunov functions

Theorem (Lyapunov's direct method)

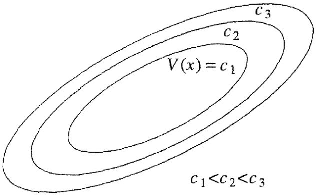
Consider $\dot{x} = f(x)$ *where* f *is locally Lipschitz and* $f(0) = 0$. *Suppose there exists* $V \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$ *such that*

- V *is positive-definite, i.e.,* $V(x) \geq 0$ *and* $V(x) = 0 \iff x = 0$,
- $\dot{V}$ *is negative-definite, i.e.,* $\nabla V(x)^\mathsf{T} f(x) \leq 0$ *and* $\nabla V(x)^\mathsf{T} f(x) = 0 \iff x = 0$.

Then $\bar{x} = 0$ *is locally asymptotically stable. If in addition*

- V *is radially unbounded, i.e.,* $V(x) \to \infty$ *as* $\|x\| \to \infty$,

then $\bar{x} = 0$ *is globally asymptotically stable.*



If the "energy" $V(x)$ is decreasing everywhere along trajectories, then $V(x) \to 0$ and thus $x \to 0$.

The existence of a Lyapunov function is a sufficient condition or *certificate* for stability. Pointwise Lyapunov inequalities are generally less cumbersome to work with than limits.

Converse Lyapunov theorems

The existence of a Lyapunov function is also necessary for stability.

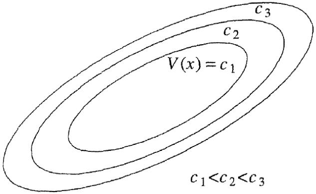
Theorem (Converse Lyapunov theorem)

Consider $\dot{x} = f(x)$ where f is locally Lipschitz. Suppose $\bar{x} = 0$ is a locally asymptotically stable equilibrium with region of attraction $\mathcal{A} \subset \mathbb{R}^n$. Then there exists $V \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$ such that

- *V is positive-definite on $\mathcal{A}$,*
- *$\dot{V}$ is negative-definite on $\mathcal{A}$,*
- *$V(x) \to \infty$ as $x \to \partial\mathcal{A}$ (the boundary of $\mathcal{A}$),*
- *$\{x \mid V(x) \leq c\}$ is a compact subset of $\mathcal{A}$ for any $c > 0$.*

If $\bar{x} = 0$ is globally asymptotically stable, i.e., $\mathcal{A} = \mathbb{R}^n$, then

- *V is radially unbounded.*



If the "energy" $V(x)$ is decreasing everywhere along trajectories, then $V(x) \to 0$ and thus $x \to 0$.

Agenda

1. Context and course goals
2. Stability and Lyapunov functions
3. Optimal control problems

Optimal control problems (continuous-time)

$$
\begin{aligned}
\underset{x,u}{\text{minimize}}\ \ & J(x,u) \coloneqq \ell_T(T, x(T)) + \int_0^T \ell(t, x(t), u(t))\, dt && \text{cost (terminal + stage)} \\
\text{subject to}\ \ & \dot{x}(t) = f(t, x(t), u(t)),\ \forall t \in [0, T] && \text{dynamical feasibility} \\
& x(t_0) = x_0,\ x(T) \in \mathcal{X}_T && \text{boundary conditions} \\
& x(t) \in \mathcal{X},\ \forall t \in [0, T] && \text{state constraints} \\
& u(t) \in \mathcal{U},\ \forall t \in [0, T] && \text{input constraints}
\end{aligned}
$$

An optimal control $u^*(t)$ for a specific initial state x_0 is an *open-loop* input. An optimal control of the form $u^*(t) = \pi^*(t, x(t))$ is a *closed-loop* input.

The stochastic and unknown model settings will be covered later on in the course.

Optimal control problems (discrete-time)

$$
\begin{aligned}
\underset{x,u}{\text{minimize}}\quad & J(x,u) := \ell_T(T, x_T) + \sum_{t=0}^{T-1} \ell(t, x_t, u_t) && \text{cost (terminal + stage)} \\
\text{subject to}\quad & x_{t+1} = f(t, x_t, u_t),\ \forall t \in \{0, 1, \ldots, T-1\} && \text{dynamical feasibility} \\
& x_0 = \bar{x}_0,\ x_T \in \mathcal{X}_T && \text{boundary conditions} \\
& x_t \in \mathcal{X},\ \forall t \in \{0, 1, \ldots, T-1\} && \text{state constraints} \\
& u_t \in \mathcal{U},\ \forall t \in \{0, 1, \ldots, T-1\} && \text{input constraints}
\end{aligned}
$$

An optimal control u_t^* for a specific initial state x_0 is an *open-loop* input. An optimal control of the form $u_t^* = \pi^*(t, x_t)$ is a *closed-loop* input.

The stochastic and unknown model settings will be covered later on in the course.

Example: Finite-horizon linear quadratic regulator (LQR)

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & x(T)^{\mathsf{T}} Q_T x(T) + \int_0^T \big(x(t)^{\mathsf{T}} Q(t) x(t) + u(t)^{\mathsf{T}} R(t) u(t) \big)\, dt && \text{cost} \\
\text{subject to} \quad & \dot{x}(t) = A(t)x(t) + B(t)u(t), \ \forall t \in [0, T] && \text{dynamical feasibility} \\
& x(0) = x_0 && \text{initial condition}
\end{aligned}
$$

For linear dynamics and a quadratic cost, we can derive *the* optimal feedback law $u^*(t) = K(t)x(t)$, which is also linear.

Example: Infinite-horizon linear quadratic regulator (LQR)

$$\underset{x,u}{\text{minimize}} \quad \int_0^\infty \big(x(t)^\mathsf{T} Q x(t) + u(t)^\mathsf{T} R u(t)\big)\, dt \qquad \text{cost}$$

$$\begin{aligned} \text{subject to} \quad & \dot{x}(t) = Ax(t) + Bu(t), \ \forall t \in [0, \infty) && \text{dynamical feasibility} \\ & x(t_0) = x_0 && \text{initial condition} \end{aligned}$$

For LTI dynamics and a time-invariant quadratic cost, we can derive *the* optimal feedback law $u^*(t) = Kx(t)$, which is also LTI.

The closed-loop system must converge to zero (i.e., be asymptotically stable) to ensure the infinite-horizon cost is well-defined.

The cost function $J(x^*, u^*)$ is a Lyapunov function for the closed-loop dynamics!

Next class

Nonlinear optimization theory
(for unconstrained and constrained problems)