AA203
Optimal and Learning-based Control
Course overview; control, stability, performance metrics
Course mechanics

Teaching team:
• Instructors: Ed Schmerling (OH: W 11am-12pm; Project OH: W 4:30-5:30pm)  
  James Harrison (OH: M 10-11am; Project OH: Th 2-3pm)
• CAs: Matt Tsao and Spencer M. Richards (OH: Tu 4-6pm, Th 8:30-10:30am)

Logistics:
• Class info, lectures, and homework assignments on class web page:  
  http://asl.stanford.edu/aa203/
• Forum: http://piazza.com/stanford/spring2021/aa203
• For urgent questions: aa203-spr2021-staff@lists.stanford.edu
Course requirements

• Homework: there will be a total of four problem sets
• Homework submissions: https://www.gradescope.com/courses/257531
• Final project (details on the course website)
• Grading:
  • homework 60% (15% per HW)
  • final project 40%
Course material

• Course notes: a set of course notes will be provided covering all the content presented in the lectures

• Recitations: Friday lecture sessions (F 10:30-11:50AM, weeks 2—5) 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

- **Strong** familiarity with calculus (e.g., CME100)
- **Strong** familiarity with linear algebra (e.g., EE263 or CME200)
- Familiarity with 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 gauge preparedness.
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

Thermostat setting + Gas flow rate → Controller → Furnace/home

- Measured temperature

Thermometer → Room temperature
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

Control
  Feedback control
  Adaptive control

Optimal and learning control
  Adaptive optimal control
  Model-free RL
  Model-based RL

Open-loop
  Indirect methods
  Direct methods

MPC
  Closed-loop

Closed-loop
  DP
  HJB / HJI

Adaptive control

Feedback 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

\[
\begin{align*}
\dot{x}_1(t) &= f_1(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t) \\
\dot{x}_2(t) &= f_2(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t) \\
&\quad \vdots \\
\dot{x}_n(t) &= f_n(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t)
\end{align*}
\]

Where

- \(x_1(t), x_2(t), \ldots, x_n(t)\) are the state variables
- \(u_1(t), u_2(t), \ldots, u_m(t)\) are the control inputs
Mathematical model

In compact form

\[ \dot{x}(t) = f(x(t), u(t), t) \]

• a history of control input values during the interval \([t_0, t_f]\) is called a control history and is denoted by \(u\)

• a history of state values during the interval \([t_0, t_f]\) is called a state trajectory and is denoted by \(x\)
Illustrative example

- Double integrator: point mass under controlled acceleration

\[
\dot{s}(t) = a(t)
\]

\[
\begin{bmatrix}
\dot{s} \\
\dot{v}
\end{bmatrix} =
\begin{bmatrix}
v \\
a
\end{bmatrix}
\]
Example system

• Double integrator: point mass under controlled acceleration

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

\[
\dot{x}(t) = A x(t) + B u(t)
\]

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

Let’s drive from \([5, 0]^T\) to \([0, 0]^T\).

Proposal: use a linear feedback control law.

\[
a = -k_p s - k_d v
\]

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

\[
\left( \begin{bmatrix}
\dot{s} \\
\dot{v}
\end{bmatrix} = (A - BK)x(t)
\right)
\]
Analyzing stability

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

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

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

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

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

where \( \lambda = -k_d / 2 \), if \( k_d^2 - 4k_p = 0 \)
Analyzing stability

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

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

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

Im(\( \lambda \)) \( \rightarrow \) sinusoidal oscillation

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

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

system comes to a stop somewhere

system exponentially converges to 0

system oscillates

system drifts off

at least one eigenvalue has positive real part; system blows up
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

\[
\begin{align*}
\min \int_0^{t_f} \left\| x(t) \right\|_2^2 + \left\| u(t) \right\|_2^2 dt \\
\text{s.t.} \quad \dot{x}(t) &= A x(t) + B u(t) \\
\hspace{2cm} x(0) &= x_0
\end{align*}
\]
Quantifying performance

\[
\min \int_0^{t_f} \left\| x(t) \right\|_2^2 + \left\| u(t) \right\|_2^2 dt
\]

s.t. \( \dot{x}(t) = Ax(t) + Bu(t) \)
\( x(0) = x_0, \ x(t_f) = x_f \)
Quantifying performance

\[
\min \int_0^{t_f} x(t)^T Q x(t) + u(t)^T R u(t) \, dt
\]

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

\( x(0) = x_0, \quad x(t_f) = x_f \)
Quantifying performance

$$\min \int_0^{t_f} x(t)^T Q x(t) + \|u(t)\|_1 dt$$

s.t. \quad \dot{x}(t) = Ax(t) + Bu(t)

x(0) = x_0, \quad x(t_f) = x_f
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

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

- \( h \) and \( g \) are scalar functions
- \( t_f \) may be specified or free
Constraints

• initial and final conditions (boundary conditions)
  \[ x(t_0) = x_0, \quad x(t_f) = x_f \]

• constraints on state trajectories
  \[ X \leq x(t) \leq \bar{X} \]

• control authority
  \[ U \leq u(t) \leq \bar{U} \]

• 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* \( u^* \) which causes the system

\[
\dot{x}(t) = f(x(t), u(t), t)
\]

to follow an *admissible trajectory* \( x^* \) that minimizes the performance measure

\[
J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) \, dt
\]
Optimal control problem

Comments:
- minimizer \((x^*, u^*)\) 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 \(\min_{u} -J\)
Form of optimal control

1. if $u^* = \pi(x(t), t)$, then $\pi$ is called optimal control law or optimal policy (closed-loop)
   • important example: $\pi(x(t), t) = F \cdot x(t)$

2. if $u^* = e(x(t_0), t)$, then the optimal control is open-loop
   • optimal only for a particular initial state value
Discrete-time formulation

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

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

- **Decision-making problem:**

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

Extension to stochastic setting will be covered later in the course
Next class

Introduction to learning;
System identification and adaptive control