AA203
Optimal and Learning-based Control

Course overview, nonlinear optimization

Stanford University

Autonomous Systems Lab

Course mechanics

Teaching team:

- Instructor: Marco Pavone
- CAs: James Harrison and Jonathan Lacotte
- Collaborators: Riccardo Bonalli (Stanford) and Roberto Calandra (FAIR)

Logistics:

- Class info, lectures, and homework assignments on class web page: http://asl.stanford.edu/aa203/
- Forum: https://piazza.com/

Course requirements

- Weekly homework, due Wednesday
- Midterm exam (05/02)
- Final project (more details later)
- Grading:
  - homework 30%
  - midterm 30%
  - final project 35%
  - grading quality 5%

Prerequisites

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

Outline

1. Problem formulation and course goals
2. Non-linear optimization
3. Computational methods

Outline

1. Problem formulation and course goals
2. Non-linear optimization
3. Computational methods

Problem formulation

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

Mathematical model

$$
\begin{aligned}
\dot{x}_1(t) &= f_1(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t) \\
\dot{x}_2(t) &= f_2(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t) \\
&\vdots \qquad\qquad \vdots \\
\dot{x}_n(t) &= f_n(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t)
\end{aligned}
$$

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

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

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

Constraints

- initial and final conditions (boundary conditions)

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

- 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

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 and g are scalar functions
- t_f may be specified or free

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

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

Very general problem formulation!

Optimal control problem

Comments:

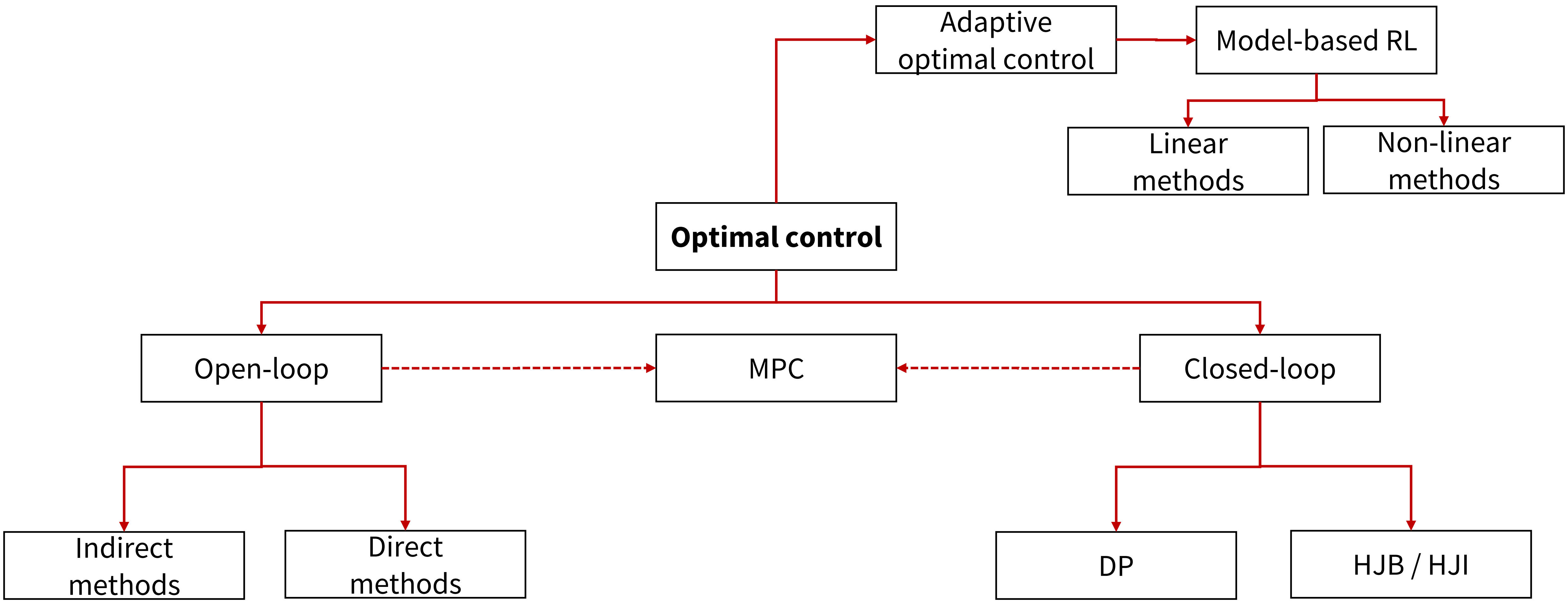
- minimizer $(\mathbf{u}^*, \mathbf{x}^*)$ called optimal trajectory-control pair
- existence: in general, not guaranteed
- uniqueness: optimal control may not be unique
- minimality: we are seeking a global minimum
- for maximization, we rewrite the problem as $\min_{\mathbf{u}} -J$

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: $\pi(\mathbf{x}(\mathrm{t}), t) = F\,\mathbf{x}(t)$

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

Course overview



Course goals

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

Outline

1. Problem formulation and course goals
2. Non-linear optimization
3. Computational methods

Non-linear optimization

Unconstrained non-linear program

$$\min_{\mathbf{x} \in \mathbb{R}^n} \; f(\mathbf{x})$$

- f usually assumed continuously differentiable (and often twice continuously differentiable)

Local and global minima

- A vector $\mathbf{x}^*$ is said an unconstrained *local* minimum if $\exists \epsilon > 0$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \qquad \forall \mathbf{x} \,\middle|\, \|\mathbf{x} - \mathbf{x}^*\| < \epsilon$$

- A vector $\mathbf{x}^*$ is said an unconstrained *global* minimum if

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \qquad \forall \mathbf{x} \in \mathbb{R}^n$$

- $\mathbf{x}^*$ is a strict local/global minimum if the inequality is strict

Necessary conditions for optimality

Key idea: compare cost of a vector with cost of its close neighbors

- Assume $f \in C^1$, by using Taylor series expansion

$$f(\mathbf{x}^* + \Delta\mathbf{x}) - f(\mathbf{x}^*) \approx \nabla f(\mathbf{x}^*)' \Delta\mathbf{x}$$

- If $f \in C^2$

$$f(\mathbf{x}^* + \Delta\mathbf{x}) - f(\mathbf{x}^*) \approx \nabla f(\mathbf{x}^*)' \Delta\mathbf{x} + \frac{1}{2} \Delta\mathbf{x}' \nabla^2 f(\mathbf{x}^*) \Delta\mathbf{x}$$

Necessary conditions for optimality

- We expect that if $\mathbf{x}^*$ is an unconstrained local minimum, the first order cost variation due to a small variation $\Delta\mathbf{x}$ is nonnegative, i.e.,

$$\nabla f(\mathbf{x}^*)'\Delta\mathbf{x} = \sum_{i=1}^{n} \frac{\partial f(\mathbf{x}^*)}{\partial x_i}\Delta x_i \geq 0$$

- By taking $\Delta\mathbf{x}$ to be positive and negative multiples of the unit coordinate vectors, we obtain conditions of the type

$$\frac{\partial f(\mathbf{x}^*)}{\partial x_i} \geq 0, \quad \text{and} \quad \frac{\partial f(\mathbf{x}^*)}{\partial x_i} \leq 0$$

- Equivalently we have the necessary condition

$$\boxed{\nabla f(\mathbf{x}^*) = 0} \qquad (\mathbf{x}^* \text{ is said a stationary point})$$

Necessary conditions for optimality

- Of course, also the second order cost variation due to a small variation $\Delta\mathbf{x}$ must be non-negative

$$\nabla f(\mathbf{x}^*)'\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}'\nabla^2 f(\mathbf{x}^*)\Delta\mathbf{x} \geq 0$$

- Since $\nabla f(\mathbf{x}^*)'\Delta\mathbf{x}=0$, we obtain $\Delta\mathbf{x}'\nabla^2 f(\mathbf{x}^*)'\Delta\mathbf{x} \geq 0$. Hence

$$\boxed{\nabla^2 f(\mathbf{x}^*) \text{ has to be positive semidefinite}}$$

NOC – formal

Theorem: NOC

Let $\mathbf{x}^*$ be an unconstrained local minimum of $f: \mathbb{R}^n \mapsto \mathbb{R}$ and assume that f is C^1 in an open set S containing $\mathbf{x}^*$. Then

$$\nabla f(\mathbf{x}^*) = 0 \qquad \text{(first order NOC)}$$

If in addition $f \in C^2$ within S,

$$\nabla^2 f(\mathbf{x}^*) \text{ positive semidefinite} \qquad \text{(second order NOC)}$$

SOC

- Assume that $\mathbf{x}^*$satisfies the first order NOC

$$\nabla f(\mathbf{x}^*) = 0$$

- and also assume that the second order NOC is strengthened to

$$\nabla^2 f(\mathbf{x}^*) \quad \text{positive } \textit{definite}$$

- Then, for all $\Delta\mathbf{x} \neq 0$, $\Delta\mathbf{x}'\nabla^2 f(\mathbf{x}^*)'\Delta\mathbf{x} > 0$. Hence, f tends to increase *strictly* with small excursions from $\mathbf{x}^*$, suggesting SOC…

SOC

Theorem: SOC

Let $f: \mathbb{R}^n \mapsto \mathbb{R}$ be C^2 in an open set S. Suppose that a vector $\mathbf{x}^* \in S$ satisfies the conditions

$$\nabla f(\mathbf{x}^*) = 0 \qquad \text{and} \qquad \nabla^2 f(\mathbf{x}^*) \text{ positive definite}$$

Then $\mathbf{x}^*$ is a strict unconstrained local minimum of f

Special case: convex optimization

A subset C of $\mathbb{R}^n$ is called convex if

$$\alpha \mathbf{x} + (1-\alpha)\mathbf{y} \in C, \quad \forall \mathbf{x}, \mathbf{y} \in C, \forall \alpha \in [0,\ 1]$$

Let C be convex. A function $f: C \rightarrow \mathbb{R}$ is called convex if

$$f(\alpha \mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y})$$

Let $f: C \rightarrow \mathbb{R}$ be a convex function over a convex set C

- A local minimum of f over C is also a global minimum over C. If in addition f is strictly convex, then there exists at most one global minimum of f
- If f is in C^1 and convex, and the set C is open, $\nabla f(\mathbf{x}^*) = 0$ is a necessary and sufficient condition for a vector $\mathbf{x}^* \in C$ to be a global minimum over C

Discussion

- Optimality conditions are important to filter candidates for global minima
- They often provide the basis for the design and analysis of optimization algorithms
- They can be used for sensitivity analysis

Outline

1. Problem formulation and course goals
2. Non-linear optimization
3. Computational methods

Computational methods (unconstrained case)

Key idea: iterative descent. We start at some point $\mathbf{x}^0$ (initial guess) and successively generate vectors $\mathbf{x}^1, \mathbf{x}^2, \ldots$ such that f is decreased at each iteration, i.e.,

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k), \qquad k = 0, 1, \ldots$$

The hope is to decrease f all the way to the minimum

Gradient methods

Given $\mathbf{x} \in \mathbb{R}^n$ with $\nabla f(\mathbf{x}) \neq 0$, consider the half line of vectors

$$\mathbf{x}_\alpha = \mathbf{x} - \alpha \nabla f(\mathbf{x}), \qquad \forall \alpha \geq 0$$

From first order Taylor expansion (α small)

$$f(\mathbf{x}_\alpha) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})'(\mathbf{x}_\alpha - \mathbf{x}) = f(\mathbf{x}) - \alpha \|\nabla f(\mathbf{x})\|^2$$

So for α small enough $f(\mathbf{x}_\alpha)$ is smaller than $f(\mathbf{x})$!

Gradient methods

Carrying this idea one step further, consider the half line of vectors

$$\mathbf{x}_\alpha = \mathbf{x} + \alpha\,\mathbf{d}, \qquad \forall \alpha \geq 0$$

where $\nabla f(\mathbf{x})'\mathbf{d} < \mathbf{0}$ (angle $> 90°$)

By Taylor expansion

$$f(\mathbf{x}_\alpha) \approx f(\mathbf{x}) + \alpha \nabla f(\mathbf{x})'\mathbf{d}$$

For small enough α, $f(\mathbf{x} + \alpha\mathbf{d})$ is smaller than $f(\mathbf{x})$!

Gradient methods

Broad and important class of algorithms: gradient methods

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \, \mathbf{d}^k, \qquad k = 0, 1, \ldots$$

where if $\nabla f(\mathbf{x}^{\mathrm{k}}) \neq 0$, $\mathbf{d}^{\mathrm{k}}$ is chosen so that

$$\nabla f(\mathbf{x}^k)' \mathbf{d}^k < 0$$

and the stepsize α is chosen to be positive

Gradient descent

Most often the stepsize is chosen so that

$$f(\mathbf{x}^k + \alpha^k \, \mathbf{d}^k) < f(\mathbf{x}^k), \qquad k = 0, 1, \ldots$$

and the method is called gradient descent. “Tuning” parameters:

- selecting the descent direction
- selecting the stepsize

Selecting the descent direction

General class

$$\mathbf{d}^k = -D^k \nabla f(\mathbf{x}^k), \qquad \text{where } D^k > 0$$

(Obviously, $\nabla f\left(\mathbf{x}^k\right)' \mathbf{d}^k < 0$)

Popular choices:

- Steepest descent: $D^k = I$
- Newton's method: $D^k = \left(\nabla^2 f(\mathbf{x}^k)\right)^{-1}$ provided $\nabla^2 f(\mathbf{x}^k) > 0$

Selecting the stepsize

- Minimization rule: α^k is selected such that the cost function is minimized along the direction $\mathbf{d}^k$, i.e.,

$$f(\mathbf{x}^k + \alpha^k \mathbf{d}^k) = \min_{\alpha \geq 0} f(\mathbf{x}^k + \alpha \mathbf{d}^k)$$

- Constant stepsize: $\alpha^k = s$
  - the method might diverge
  - convergence rate could be very slow

- Diminishing stepsize: $\alpha^k \rightarrow 0$ and $\sum_{k=0}^{+\infty} \alpha^k = \infty$
  - it does not guarantee descent at each iteration

Discussion

Aspects:

- convergence (to stationary points)
- termination criteria
- convergence rate

Non-derivative methods, e.g.,

- coordinate descent

Next time

Constrained non-linear optimization

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{subject to} \quad & h_i(\mathbf{x}) = 0, \qquad i = 1, \ldots, m \end{aligned}$$