Tutorial on numerical optimal control

Sébastien Kleff

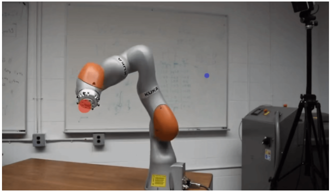
Oct. 3, 2024

Why optimal control?

Stairs climbing

https://youtu.be/v6MhPl2ICsc

Pick-and-place



https://youtu.be/ZtyCJYsGf4U

Parkour

https://youtu.be/tF4DML7FIWk

Obstacle avoidance

They all solve an **optimal control problem**

Tutorial objectives

Ideally, by the end of this tutorial :

- The relations between OC, MPC & DDP should be clear(er) to you
- You will understand words like "direct multiple shooting"
- You can implement your own MPC to control your favorite robot

To achieve these goals, I will provide

- Quick overview of Optimal Control
- Crash course on nonlinear optimization
- Tutorial using Crocoddyl and mim_solvers

Tutorial (will be) available on a dedicated repo :
`https://github.com/skleff1994/mpc_tutorial`

Tutorial plan

Tentative plan :

- Session 1 : Optimal control
- Session 2 : Numerical optimization
- Session 3 : MPC

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{1}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{1}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

Components

- State and control trajectories $x(.) : \mathbb{R}^+ \rightarrow \mathbb{R}^{n_x}, u(.) : \mathbb{R}^+ \rightarrow \mathbb{R}^{n_u}$

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{1}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

Components

- State and control trajectories $x(.) : \mathbb{R}^+ \to \mathbb{R}^{n_x}, u(.) : \mathbb{R}^+ \to \mathbb{R}^{n_u}$
- Running cost $\ell : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \to \mathbb{R}^+$ and terminal cost $\ell_T : \mathbb{R}^{n_x} \to \mathbb{R}^+$

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{1}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

Components

- State and control trajectories $x(.) : \mathbb{R}^+ \to \mathbb{R}^{n_x}, u(.) : \mathbb{R}^+ \to \mathbb{R}^{n_u}$
- Running cost $\ell : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \to \mathbb{R}^+$ and terminal cost $\ell_T : \mathbb{R}^{n_x} \to \mathbb{R}^+$
- Dynamics constraint $f : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{1}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

Components

- State and control trajectories $x(.) : \mathbb{R}^+ \to \mathbb{R}^{n_x}, u(.) : \mathbb{R}^+ \to \mathbb{R}^{n_u}$
- Running cost $\ell : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \to \mathbb{R}^+$ and terminal cost $\ell_T : \mathbb{R}^{n_x} \to \mathbb{R}^+$
- Dynamics constraint $f : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$
- Path constraints $c : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_c}$ (possibly terminal c_T)

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{1}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

Components

- State and control trajectories $x(.) : \mathbb{R}^+ \rightarrow \mathbb{R}^{n_x}, u(.) : \mathbb{R}^+ \rightarrow \mathbb{R}^{n_u}$
- Running cost $\ell : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^+$ and terminal cost $\ell_T : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^+$
- Dynamics constraint $f : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$
- Path constraints $c : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_c}$ (possibly terminal c_T)
- Initial condition $(x_0, t_0) \in \mathbb{R}^{n_x} \times \mathbb{R}^+$ and horizon $T > 0$ (can be $+\infty$)

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{1}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

Components

- State and control trajectories $x(.) : \mathbb{R}^+ \to \mathbb{R}^{n_x}, u(.) : \mathbb{R}^+ \to \mathbb{R}^{n_u}$
- Running cost $\ell : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \to \mathbb{R}^+$ and terminal cost $\ell_T : \mathbb{R}^{n_x} \to \mathbb{R}^+$
- Dynamics constraint $f : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$
- Path constraints $c : \mathbb{R}^{n_x} : \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_c}$ (possibly terminal c_T)
- Initial condition $(x_0, t_0) \in \mathbb{R}^{n_x} \times \mathbb{R}^+$ and horizon $T > 0$ (can be $+\infty$)
- Some technical assumptions

Optimal control: Manipulator example

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

A 7-DoF torque-controlled manipulator must reach an end-effector position $p^{\text{des}} \in \mathbb{R}^3$ while using mimium energy and satisfying operating constraints

Optimal control: Manipulator example

Continuous-time **Optimal Control Problem** (OCP)

$$V(x_0, t_0) = \min_{u(.)} \int_{t_0}^{T} \ell\big(x(t), u(t)\big)\,\mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\big(x(t), u(t)\big) & \forall t \in [t_0, T] \\ c\big(x(t), u(t)\big) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

A 7-DoF torque-controlled manipulator must reach an end-effector position $p^{\text{des}} \in \mathbb{R}^3$ while using mimium energy and satisfying operating constraints

- What is x ?
- What is u ?
- What is f ?
- What is ℓ ?
- What is c ?

Optimal control: Manipulator example

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

A 7-DoF torque-controlled manipulator must reach an end-effector position $p^{\text{des}} \in \mathbb{R}^3$ while using mimium energy and satisfying operating constraints

- We define the state variable $x = (q, \dot{q}) \in \mathbb{R}^{2n_q}$ and the control input $u = \tau \in \mathbb{R}^{n_q}$ with $n_q = 7$
- The dynamics constraint f is given by the robot forward dynamics

$$\underbrace{\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} = \dot{q} \\ M(q)^{-1}(\tau - h(q, \dot{q}) \end{bmatrix}}_{f(x,u)}$$

Optimal control: Manipulator example

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

A 7-DoF torque-controlled manipulator must reach an end-effector position $p^{\text{des}} \in \mathbb{R}^3$ while using mimium energy and satisfying operating constraints

- Path constraints are e.g. joint limits $[x^{\text{min}}, x^{\text{max}}]$ and torque limits $[u^{\text{min}}, u^{\text{max}}]$

$$c(x, u) = \begin{bmatrix} x - x^{\text{max}} \\ x^{\text{min}} - x \\ u - u^{\text{max}} \\ u^{\text{min}} - u \end{bmatrix}$$

Optimal control: Manipulator example

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

A 7-DoF torque-controlled manipulator must reach an end-effector position $p^{\text{des}} \in \mathbb{R}^3$ while using mimium energy and satisfying operating constraints

- Initial condition : robot at rest $x_0 = (q_0, 0)$ at $t_0 = 0$
- Cost function : penalizes the distance to the target and the energy

$$\ell(x, u) = \alpha \|u\|_2^2$$
$$\ell_T(x) = \|p(q) - p^{\text{des}}\|_2^2$$

where $\alpha > 0$ is a scalar parameter and the end-effector position $p(q)$ is given by the robot's forward kinematics

Optimal control: Solution

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

The $\min$ is the optimal cost or *value function* $V : \mathbb{R}^{n_x} \times \mathbb{R}^+ \to \mathbb{R}^+$

Optimal control: Solution

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

The $\min$ is the optimal cost or *value function* $V : \mathbb{R}^{n_x} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$

The $\arg\min$ is called the *optimal feedback control policy* $\pi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$

Optimal control: Solution

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

The $\min$ is the optimal cost or *value function* $V : \mathbb{R}^{n_x} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$

The $\arg\min$ is called the *optimal feedback control policy* $\pi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$

Very generic formulation

Optimal control: Solution

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

The $\min$ is the optimal cost or *value function* $V : \mathbb{R}^{n_x} \times \mathbb{R}^+ \to \mathbb{R}^+$

The $\arg\min$ is called the *optimal feedback control policy* $\pi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$

Very generic formulation

Well-established framework

Optimal control: Solution

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

The $\min$ is the optimal cost or *value function* $V : \mathbb{R}^{n_x} \times \mathbb{R}^+ \to \mathbb{R}^+$

The $\arg\min$ is called the *optimal feedback control policy* $\pi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$

Very generic formulation

Well-established framework

But how to solve this?

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$
$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

The decision variable $u(.) \in \left(\mathbb{R}^{n_u}\right)^{[0,T]}$ is infinite dimensional !

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \quad \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

The decision variable $u(.) \in (\mathbb{R}^{n_u})^{[0,T]}$ is infinite dimensional !

V is a *functional* characterized by the Hamilton-Jacobi-Bellman (HJB) Partial Differential Equation (PDE)

$$-\frac{\partial V(x,t)}{\partial t} = \min_{u} \left[\ell(x,u) + \mathcal{I}_c(x,u) + \frac{\partial V(x,t)}{\partial x} f(x,u)\right]$$

Optimal control: Problem definition

Continuous-time **Optimal Control Problem** (OCP)

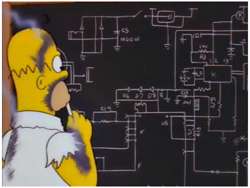
$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \ell\left(x(t), u(t)\right) \mathrm{d}t + \ell_T(x(T)) \tag{2}$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 & \\ \dot{x}(t) = f\left(x(t), u(t)\right) & \forall t \in [t_0, T] \\ c\left(x(t), u(t)\right) \leq 0 & \forall t \in [t_0, T] \end{cases}$$

The decision variable $u(.) \in \left(\mathbb{R}^{n_u}\right)^{[0,T]}$ is infinite dimensional !

V is a *functional* characterized by the Hamilton-Jacobi-Bellman (HJB) Partial Differential Equation (PDE)

$$-\frac{\partial V(x,t)}{\partial t} = \min_{u} \left[\ell(x,u) + \mathcal{I}_c(x,u) + \frac{\partial V(x,t)}{\partial x} f(x,u)\right]$$



No analytical solution except in very specific cases !

Analytic solution: Special case of LQR

Particular case : Linear-Quadratic Regulator (LQR)

When f is linear, ℓ is quadratic (no path constraints)

$$V(x_0, t_0) = \min_{u(.)} \int_{t_0}^{T} \left(x(t)^\top Q x(t) + u(t)^\top R u(t) \right) + x(T)^\top Q_T x(T) \qquad (3)$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = Ax(t) + Bu(t) \qquad \forall t \in [t_0, T] \end{cases}$$

where $Q, Q_T, R \succ 0$ and (A, B) is controllable.

Analytic solution: Special case of LQR

Particular case : Linear-Quadratic Regulator (LQR)

When f is linear, ℓ is quadratic (no path constraints)

$$V\left(x_0, t_0\right) = \min_{u(.)} \int_{t_0}^{T} \left(x(t)^\top Q x(t) + u(t)^\top R u(t)\right) + x(T)^\top Q_T x(T) \tag{3}$$

$$\text{s.t.} \quad \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = Ax(t) + Bu(t) \qquad \forall t \in [t_0, T] \end{cases}$$

where $Q, Q_T, R \succ 0$ and (A, B) is controllable.

The value function V is quadratic, i.e. $V(x,t) = x^\top P(t) x$ where $P(t) \succ 0$ is the solution the Riccati differential equation

$$-\dot{P}(t) = A^\top P(t) + P(t)A - P(t)BR^{-1}B^\top P(t) + Q \tag{4}$$

Analytic solution: Special case of LQR

Particular case : Linear-Quadratic Regulator (LQR)

When f is linear, ℓ is quadratic (no path constraints)

$$V(x_0, t_0) = \min_{u(.)} \int_{t_0}^{T} \left(x(t)^\top Q x(t) + u(t)^\top R u(t) \right) + x(T)^\top Q_T x(T) \quad (3)$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = Ax(t) + Bu(t) \qquad \forall t \in [t_0, T] \end{cases}$$

where $Q, Q_T, R \succ 0$ and (A, B) is controllable.

The value function V is quadratic, i.e. $V(x,t) = x^\top P(t) x$ where $P(t) \succ 0$ is the solution the Riccati differential equation

$$-\dot{P}(t) = A^\top P(t) + P(t)A - P(t)BR^{-1}B^\top P(t) + Q \quad (4)$$

The optimal policy is linear $\pi(x(t)) = K(t)x(t)$ where $K(t) = -R^{-1}B^\top P(t)$

Analytic solution: Special case of LQR

Particular case : Linear-Quadratic Regulator (LQR)

When f is linear, ℓ is quadratic (no path constraints)

$$V(x_0, t_0) = \min_{u(.)} \int_{t_0}^{T} \left(x(t)^\top Q x(t) + u(t)^\top R u(t) \right) + x(T)^\top Q_T x(T) \quad (3)$$

$$\text{s.t.} \begin{cases} x(t_0) = x_0 \\ \dot{x}(t) = Ax(t) + Bu(t) \qquad \forall t \in [t_0, T] \end{cases}$$

where $Q, Q_T, R \succ 0$ and (A, B) is controllable.

The value function V is quadratic, i.e. $V(x,t) = x^\top P(t) x$ where $P(t) \succ 0$ is the solution the Riccati differential equation

$$-\dot{P}(t) = A^\top P(t) + P(t)A - P(t)BR^{-1}B^\top P(t) + Q \quad (4)$$

The optimal policy is linear $\pi(x(t)) = K(t)x(t)$ where $K(t) = -R^{-1}B^\top P(t)$

In the general case, HJB must be solved numerically

Numerical solution: Simple Pendulum

Example : Pendulum swing-up task

State $x = (\theta, \dot{\theta})$, control $u = \tau$ and dynamics model f

$$ml^2\ddot{\theta} + mg\sin\theta = \tau \tag{5}$$

Running cost $\ell(x, u) = \alpha\|u\|^2$ and terminal cost $\ell_T(x) = \|x\|^2$

Numerical solution: Simple Pendulum

Example : Pendulum swing-up task

State $x = (\theta, \dot{\theta})$, control $u = \tau$ and dynamics model f

$$ml^2\ddot{\theta} + mg\sin\theta = \tau \tag{5}$$

Running cost $\ell(x, u) = \alpha\|u\|^2$ and terminal cost $\ell_T(x) = \|x\|^2$

This is not LQR (f is nonlinear) so we must solve the HJB PDE numerically

Numerical solution: Simple Pendulum

Example : Pendulum swing-up task

State $x = (\theta, \dot{\theta})$, control $u = \tau$ and dynamics model f

$$ml^2\ddot{\theta} + mg\sin\theta = \tau \tag{5}$$

Running cost $\ell(x, u) = \alpha\|u\|^2$ and terminal cost $\ell_T(x) = \|x\|^2$

This is not LQR (f is nonlinear) so we must solve the HJB PDE numerically

We **discretize** the state and control spaces into finite meshes

We solve the PDE numerically to compute V explicitly for every $(\theta, \dot{\theta})$

Check-out the pendulum example and play with it : `pendulum_bellman.py`

Numerical solution: Simple Pendulum

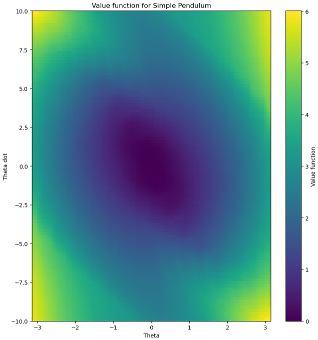
Example : Pendulum swing-up task

State $x = (\theta, \dot{\theta})$, control $u = \tau$ and dynamics model f

$$ml^2\ddot{\theta} + mg\sin\theta = \tau \qquad (5)$$

Running cost $\ell(x, u) = \alpha\|u\|^2$ and terminal cost $\ell_T(x) = \|x\|^2$

This is not LQR (f is nonlinear) so we must solve the HJB PDE numerically



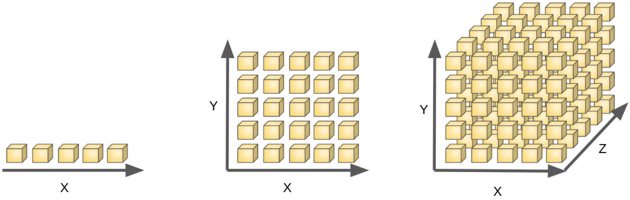
We **discretize** the state and control spaces into finite meshes

We solve the PDE numerically to compute V explicitly for every $(\theta, \dot{\theta})$

Check-out the pendulum example and play with it : `pendulum_bellman.py`

Numerical solution: Curse of dimensionality

Major problem : the number of points N required to maintain the same sampling density increases **exponentially** with the state space dimension n_x



100 points per dimension with $n_x = 2$: $N = 10^4$ points

100 points per dimension with $n_x = 6$: $N = 10^8$ points

Our 7-DoF torque-controlled manipulator has $n_x = 14$...

Computing V explicitly is not tractable if $n_x \geq 4$ or 5

Recap

Analytic resolution of HJB PDE : *"Explicit formula for V"*

- Exact *global* solution (i.e. compute $V(x)$ for all x)
- Feedback (closed-loop) policy $\pi(x)$
- Only works in very specific cases (e.g. LQR)

Recap

Analytic resolution of HJB PDE : *"Explicit formula for V"*

- Exact *global* solution (i.e. compute $V(x)$ for all x)
- Feedback (closed-loop) policy $\pi(x)$
- Only works in very specific cases (e.g. LQR)

Numerical resolution of HJB PDE : *"Integrate the HJB PDE"*

- Approximate global solution
- Feedback (closed-loop) policy $\pi(x)$
- Curse of dimensionality

Recap

Analytic resolution of HJB PDE : *"Explicit formula for V"*

- Exact *global* solution (i.e. compute $V(x)$ for all x)
- Feedback (closed-loop) policy $\pi(x)$
- Only works in very specific cases (e.g. LQR)

Numerical resolution of HJB PDE : *"Integrate the HJB PDE"*

- Approximate global solution
- Feedback (closed-loop) policy $\pi(x)$
- Curse of dimensionality

Luckily there is an alternative : direct optimal control

Direct Optimal Control

1. "First discretize" : Transform OCP into Nonlinear Program
2. "Then optimize" : Solve Nonlinear Program

Direct Optimal Control

1. "First discretize" : Transform OCP into Nonlinear Program
2. "Then optimize" : Solve Nonlinear Program

- Can deal with large, difficult problems

Direct Optimal Control

1. "First discretize" : Transform OCP into Nonlinear Program
2. "Then optimize" : Solve Nonlinear Program

- Can deal with large, difficult problems
- Many solvers available

Direct Optimal Control

1. "First discretize" : Transform OCP into Nonlinear Program
2. "Then optimize" : Solve Nonlinear Program

- Can deal with large, difficult problems
- Many solvers available
- Widely used in robotics

Direct Optimal Control

1. "First discretize" : Transform OCP into Nonlinear Program
2. "Then optimize" : Solve Nonlinear Program

- Can deal with large, difficult problems
- Many solvers available
- Widely used in robotics
- Relies on well-established numerical optimization tools

Direct Optimal Control

1. "First discretize" : Transform OCP into Nonlinear Program
2. "Then optimize" : Solve Nonlinear Program

- Can deal with large, difficult problems
- Many solvers available
- Widely used in robotics
- Relies on well-established numerical optimization tools
- Local solutions only (only valid around some given $x(t)$)

Direct Optimal Control

1. "First discretize" : Transform OCP into Nonlinear Program
2. "Then optimize" : Solve Nonlinear Program

- Can deal with large, difficult problems
- Many solvers available
- Widely used in robotics
- Relies on well-established numerical optimization tools
- Local solutions only (only valid around some given $x(t)$)
- Control trajectories $u(t)$ (open-loop policy)*

***Spoiler** : *MPC will essentially use direct optimal control to approximate the closed-loop policy $\pi(x)$ through repeated open-loop solutions $u(t)$*

Direct approach: "First discretize"

Transcription : Parametrize the infinite-dimensional OCP (2)

Direct approach: "First discretize"

Transcription : Parametrize the infinite-dimensional OCP (2)

1. **Discretization** of the time horizon $t_0, ..., t_N$ with $t_k = k\delta$ for $k \in [0, N]$

Direct approach: "First discretize"

Transcription : Parametrize the infinite-dimensional OCP (2)

1. **Discretization** of the time horizon $t_0, ..., t_N$ with $t_k = k\delta$ for $k \in [0, N]$
2. **Parametrization** e.g. zero-order hold $u(t) = u(t_k)$ for $t \in [t_k, t_{k+1})$

Direct approach: "First discretize"

Transcription : Parametrize the infinite-dimensional OCP (2)

1. **Discretization** of the time horizon $t_0, ..., t_N$ with $t_k = k\delta$ for $k \in [0, N]$
2. **Parametrization** e.g. zero-order hold $u(t) = u(t_k)$ for $t \in [t_k, t_{k+1})$
3. **Integration** of the state dynamics $x(.)$, e.g. using Euler and midpoint rule

$$x_{k+1} = x_k + \underbrace{F(x_k, u_k)}_{\text{continuous-time dynamics}} \delta \tag{6}$$

$$\ell(x_k, u_k) = \underbrace{L(x_k, u_k)}_{\text{continuous-time cost}} \delta \tag{7}$$

Direct approach: "First discretize"

Transcription : Parametrize the infinite-dimensional OCP (2)

1. **Discretization** of the time horizon $t_0, ..., t_N$ with $t_k = k\delta$ for $k \in [0, N]$
2. **Parametrization** e.g. zero-order hold $u(t) = u(t_k)$ for $t \in [t_k, t_{k+1})$
3. **Integration** of the state dynamics $x(.)$, e.g. using Euler and midpoint rule

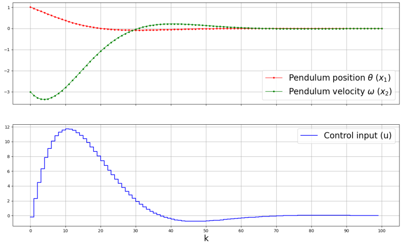
$$x_{k+1} = x_k + \underbrace{F(x_k, u_k)}_{\text{continuous-time dynamics}} \delta \tag{6}$$

$$\ell(x_k, u_k) = \underbrace{L(x_k, u_k)}_{\text{continuous-time cost}} \delta \tag{7}$$

Pendulum with semi-explicit Euler

$$\dot{\theta}_{k+1} = \dot{\theta}_k - \overbrace{\left(\frac{-g\sin(\theta)}{l} + u\right)}^{\text{continuous-time acceleration}} \delta$$

$$\theta_{k+1} = \theta_k + \delta\dot{\theta}_{k+1}$$



Direct approach: "First discretize"

This leads to a *discrete-time OCP* with *finite-dimensional* decision variables

$$\min_{u_0,\ldots,u_{T-1}} \sum_{k=0}^{T-1} \ell(x_k, u_k) + \ell_T(x_T) \tag{8}$$

$$\text{s.t.} \begin{cases} x_0 = \hat{x} \\ x_{k+1} = f(x_k, u_k) \\ c(x_k, u_k) \leq 0 \end{cases}$$

Remarks

Direct approach: "First discretize"

This leads to a *discrete-time OCP* with *finite-dimensional* decision variables

$$\min_{u_0,\dots,u_{T-1}} \sum_{k=0}^{T-1} \ell(x_k, u_k) + \ell_T(x_T) \tag{8}$$

$$\text{s.t.} \begin{cases} x_0 = \hat{x} \\ x_{k+1} = f(x_k, u_k) \\ c\,(x_k, u_k) \leq 0 \end{cases}$$

Remarks

- Discrete-time problems are much easier to study (see Bertsekas)

Direct approach: "First discretize"

This leads to a *discrete-time OCP* with *finite-dimensional* decision variables

$$\min_{u_0,\ldots,u_{T-1}} \sum_{k=0}^{T-1} \ell(x_k, u_k) + \ell_T(x_T) \tag{8}$$
$$\text{s.t.} \begin{cases} x_0 = \hat{x} \\ x_{k+1} = f(x_k, u_k) \\ c(x_k, u_k) \leq 0 \end{cases}$$

Remarks

- Discrete-time problems are much easier to study (see Bertsekas)
- Most of continuous-time optimal control theory applies in discrete-time

Direct approach: "First discretize"

This leads to a *discrete-time OCP* with *finite-dimensional* decision variables

$$\min_{u_0,\dots,u_{T-1}} \sum_{k=0}^{T-1} \ell(x_k,u_k) + \ell_T(x_T) \tag{8}$$

$$\text{s.t.} \quad \begin{cases} x_0 = \hat{x} \\ x_{k+1} = f(x_k,u_k) \\ c\,(x_k,u_k) \le 0 \end{cases}$$

Remarks

- Discrete-time problems are much easier to study (see Bertsekas)
- Most of continuous-time optimal control theory applies in discrete-time
- For instance, the Bellman equation is the discrete-time equivalent of HJB

$$V_j(x) = \min_u \; V_{j+1}\big(f(x,u)\big) + \ell(x,u) + \mathcal{I}_c(x,u) \tag{9}$$

where V_j is the optimal cost-to-go at stage j

Discrete-time OCP: Special case of LQR

Particular case : Linear-Quadratic Regulator (LQR)

When f is linear, ℓ is quadratic (no path constraints)

$$V_0\left(x_0\right) = \min_{u_0,\ldots,u_{T-1}} \sum_{k=0}^{T-1} \left(x_k^\top Q x_k + u_k^\top R u_k\right) + x_T^\top Q_T x_T \tag{10}$$

$$\text{s.t.} \begin{cases} x_0 = \hat{x} \\ x_{k+1} = A x_k + B u_k \qquad \forall k \in \{0, ..., N-1\} \end{cases}$$

where $Q, Q_T, R \succ 0$ and (A, B) is controllable.

Discrete-time OCP: Special case of LQR

Particular case : Linear-Quadratic Regulator (LQR)

When f is linear, ℓ is quadratic (no path constraints)

$$V_0\left(x_0\right) = \min_{u_0,\ldots,u_{T-1}} \sum_{k=0}^{T-1} \left(x_k^\top Q x_k + u_k^\top R u_k\right) + x_T^\top Q_T x_T \tag{10}$$

$$\text{s.t.} \begin{cases} x_0 = \hat{x} \\ x_{k+1} = A x_k + B u_k \qquad \forall k \in \{0, \ldots, N-1\} \end{cases}$$

where $Q, Q_T, R \succ 0$ and (A, B) is controllable.

The value function V is quadratic, i.e. $V_k(x) = x^\top P_k x$ where $P_k \succ 0$ is the solution the Riccati differential equation

$$P_k = A^\top P_k A - (A^\top P_k B)(R + B^\top P_k B)^{-1} B^\top P_k A + Q \tag{11}$$

Discrete-time OCP: Special case of LQR

Particular case : Linear-Quadratic Regulator (LQR)

When f is linear, ℓ is quadratic (no path constraints)

$$V_0\left(x_0\right) = \min_{u_0,\ldots,u_{T-1}} \sum_{k=0}^{T-1} \left(x_k^\top Q x_k + u_k^\top R u_k\right) + x_T^\top Q_T x_T \tag{10}$$

$$\text{s.t.} \quad \begin{cases} x_0 = \hat{x} \\ x_{k+1} = A x_k + B u_k \qquad \forall k \in \{0, \ldots, N-1\} \end{cases}$$

where $Q, Q_T, R \succ 0$ and (A, B) is controllable.

The value function V is quadratic, i.e. $V_k(x) = x^\top P_k x$ where $P_k \succ 0$ is the solution the Riccati differential equation

$$P_k = A^\top P_k A - (A^\top P_k B)(R + B^\top P_k B)^{-1} B^\top P_k A + Q \tag{11}$$

The optimal policy is linear $\pi_k(x) = K_k x$ where $K_k = -R^{-1} B^\top P_k$

Discrete-time OCP: Special case of LQR

Particular case : Linear-Quadratic Regulator (LQR)

When f is linear, ℓ is quadratic (no path constraints)

$$V_0\left(x_0\right) = \min_{u_0,\ldots,u_{T-1}} \sum_{k=0}^{T-1} \left(x_k^\top Q x_k + u_k^\top R u_k\right) + x_T^\top Q_T x_T \tag{10}$$

$$\text{s.t.} \begin{cases} x_0 = \hat{x} \\ x_{k+1} = A x_k + B u_k \qquad \forall k \in \{0, ..., N-1\} \end{cases}$$

where $Q, Q_T, R \succ 0$ and (A, B) is controllable.

The value function V is quadratic, i.e. $V_k(x) = x^\top P_k x$ where $P_k \succ 0$ is the solution the Riccati differential equation

$$P_k = A^\top P_k A - (A^\top P_k B)(R + B^\top P_k B)^{-1} B^\top P_k A + Q \tag{11}$$

The optimal policy is linear $\pi_k(x) = K_k x$ where $K_k = -R^{-1} B^\top P_k$

In the general case, there is no analytic solution for V, π

Direct approach: "Then optimize"

The discrete-time OCP

$$V_0(x_0) = \min_{u_0,\dots,u_{T-1}} \sum_{k=0}^{T-1} \ell(x_k, u_k) + \ell_T(x_T) \tag{8}$$

$$\text{s.t.} \begin{cases} x_0 = \hat{x} \\ x_{k+1} = f(x_k, u_k) \\ c\left(x_k, u_k\right) \leq 0 \end{cases}$$

can in theory be solved **globally** in the general case by solving Bellman's equation, but it suffers from the curse of dimensionality (like HJB).

Direct approach: "Then optimize"

The discrete-time OCP

$$V_0(x_0) = \min_{u_0,\dots,u_{T-1}} \sum_{k=0}^{T-1} \ell(x_k, u_k) + \ell_T(x_T) \tag{8}$$

$$\text{s.t.} \begin{cases} x_0 = \hat{x} \\ x_{k+1} = f(x_k, u_k) \\ c\left(x_k, u_k\right) \leq 0 \end{cases}$$

can in theory be solved **globally** in the general case by solving Bellman's equation, but it suffers from the curse of dimensionality (like HJB).

Instead, we seek a **local** solution by solving (8) as a Nonlinear Program (NLP)

$$\min_{z} \ L(z) \tag{12}$$

$$\text{s.t. } G(z) \leq 0$$

where $z \in \mathbb{R}^n$, $L : \mathbb{R}^n \to \mathbb{R}^+$ and $G : \mathbb{R}^n \to \mathbb{R}^{n_c}$.

Direct approach: "Then optimize"

The discrete-time OCP

$$V_0(x_0) = \min_{u_0,\dots,u_{T-1}} \sum_{k=0}^{T-1} \ell(x_k, u_k) + \ell_T(x_T) \tag{8}$$

$$\text{s.t.} \quad \begin{cases} x_0 = \hat{x} \\ x_{k+1} = f(x_k, u_k) \\ c\left(x_k, u_k\right) \leq 0 \end{cases}$$

can in theory be solved **globally** in the general case by solving Bellman's equation, but it suffers from the curse of dimensionality (like HJB).

Instead, we seek a **local** solution by solving (8) as a Nonlinear Program (NLP)

$$\min_z \; L(z) \tag{12}$$

$$\text{s.t.} \; G(z) \leq 0$$

where $z \in \mathbb{R}^n$, $L : \mathbb{R}^n \to \mathbb{R}^+$ and $G : \mathbb{R}^n \to \mathbb{R}^{n_c}$.

This is a standard nonlinear optimization problem that can be solved using textbook numerical optimization

Recap

- Optimal control is a generic framework
- OCPs are challenging to solve globally
- We can seek local solutions instead
- This requires to transform the original OCP into an NLP

Next time

Session 2 will focus on the NLP resolution

- What are the main techniques to solve a generic NLP
- How our NLP has an special structure
- How this structure can be exploited to solve it efficiently

Session 3 will focus on MPC implementation and introduction of the existing tools (Crocoddyl, mim_solvers)