reference

Intro to Optimal Control

A (Biased) Brief Introduction

Haotian Yang
younght@qq.com

College of Automation
Chongqing University

Created on April 16, 2025

Outline

- What is Optimal Control?
- Why Optimal Control?
- Key Concepts
  - System dynamics and variables
  - Cost functions and constraints
- Common Methods
  - Classical approaches
  - Numerical techniques
- Case Study: Newton's Method KKT conditions
- Examples
  - Aerospace applications
  - Rocket landing and quadrotors
- Summary

What is Optimal Control?

What is Optimal Control?

- Optimal control deals with finding a control strategy for a dynamic system over a period of time to achieve a specific objective optimally.
- What does "optimal" mean?
    - Minimizing a cost function (e.g., time, fuel consumption, error).
    - Maximizing a reward function (e.g., profit, distance covered, product yield).
- It essentially answers the question: "How should I operate my system to achieve a goal in the *best* possible way?"
- Requires balancing desired performance against system limitations and constraints.

Why Optimal Control?

Why Study Optimal Control? Applications

Optimal control principles are fundamental in many fields:

- **Aerospace Engineering:** Minimum-fuel or minimum-time spacecraft trajectories, missile guidance, aircraft flight control.
- **Robotics:** Path planning for robot arms or mobile robots (e.g., fastest path, least energy consumption).
- **Chemical Engineering:** Optimizing reactor conditions to maximize product yield or minimize operating costs.
- **Economics & Finance:** Optimal resource allocation, investment strategies, inventory management.
- **Automotive:** Autonomous driving (path planning, speed control), hybrid vehicle energy management.
- **Power Systems:** Economic dispatch, load frequency control.

Key Concepts

Key Components of an Optimal Control Problem

- **System Dynamics:** Mathematical model describing how the system's state evolves over time. Often represented by differential equations:

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

  where x is the state vector, u is the control input vector
- **Control Inputs (u):** The variables we can manipulate or choose to influence the system's behavior.
- **State Variables (x):** Variables that characterize the condition or configuration of the system at any given time.
- **Cost Function (Objective Function, J):** A scalar value quantifying the performance. We aim to minimize (or maximize) this. A common form is:

$$J = \Phi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t)\, dt$$

Conceptual Example: Driving a Car

Imagine driving from Point A to Point B:

- **System Dynamics:** Physics of the car (position, velocity, acceleration based on engine/braking forces).
- **State Variables (x):** Position, velocity, fuel level.
- **Control Inputs (u):** Steering angle, accelerator position, brake pressure.
- **Possible Objectives (Cost Functions J):**
  - Minimize travel time: $J = t_f - t_0$.
  - Minimize fuel consumption: $J = \int_{t_0}^{t_f} \text{fuel\_rate}(u(t))\,dt$.
  - Minimize discomfort (e.g., jerky movements): $J = \int_{t_0}^{t_f} (\text{acceleration}^2 + \text{jerk}^2)\,dt$.
- **Constraints:** Speed limits, road boundaries, maximum acceleration/braking, car cannot go in reverse (usually!).

Optimal control finds the sequence of steering/accelerator/brake actions ($u(t)$) that achieves the chosen objective best, given the car's dynamics and constraints.

Common Methods

How are Optimal Control Problems Solved?

Finding the optimal control $u^*(t)$ often involves advanced mathematical techniques:

- **Calculus of Variations:** The historical foundation, dealing with minimizing functionals (functions of functions).
- **Pontryagin's Maximum Principle (PMP):**
  - Provides *necessary conditions* for optimality.
  - Introduces costate variables and a Hamiltonian function.
  - Often leads to solving a two-point boundary value problem (TPBVP), which can be challenging.
- **Dynamic Programming (DP):**
  - Based on Bellman's Principle of Optimality ("An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.")
  - Leads to the Hamilton-Jacobi-Bellman (HJB) equation (a

Case Study

Newton's Methods

Newton's Method: Solving a system of nonlinear equations $F(x) = 0$

- We **can not** solve arbitrary "*nonlinear equations*" very easily.
- We **can** solve "*linear equations*" using the techniques from linear algebra.
- We linearize the system of "*nonlinear equations*" with a first-order Taylor and solve the obtained "*linear system*", and then iterate.

(0) *k=0,*

(1) *Linear approximation at point* x_k:

$$y = F(x_k) + \nabla F(x_k)^T (x - x_k)$$

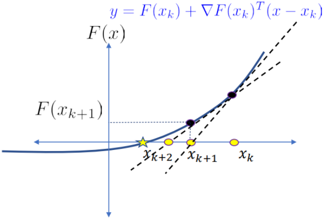
(2) *Solve linear system:*

$$y = 0 \longrightarrow F(x_k) + \nabla F(x_k)'(x - x_k) = 0$$

$$\Rightarrow (x - x_k) = -(\nabla F(x_k))^{-1} F(x_k)$$

$$\Rightarrow \boxed{x_{k+1} = x_k - (\nabla F(x_k))^{-1} F(x_k)}$$

(3) *Go to Step* (1)



KKT condition

$$
\begin{aligned}
\min_{x_{1:N},\, u_{1:N-1}} \quad & \left[\sum_{i=1}^{N-1} \ell(x_i, u_i) \right] + \ell_N(x_N) \\
\text{s.t.} \quad & x_1 = x_{\text{IC}} \\
& x_{k+1} = f(x_k, u_k), \quad \text{for } k = 1, 2, \ldots, N-1
\end{aligned}
$$

KKT condition

We are going to solve an equality-constrained optimization problem:

$$\min_{x} \quad f(x)$$
$$\text{s.t.} \quad c(x) = 0$$

The Lagrangian of the problem is defined as:

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^T c(x)$$

The corresponding KKT conditions for optimality are:

$$\nabla_x \mathcal{L} = \nabla_x f(x) + \left(\frac{\partial c}{\partial x}\right)^T \lambda = 0$$
$$c(x) = 0$$

This is essentially a root-finding problem. We aim to solve for

$$z = \begin{bmatrix} x \\ \lambda \end{bmatrix}$$

such that the KKT conditions are satisfied.

How to solve?

*B*ut, Do we really need to solve these complex math problems?

How to solve?

Julia

- High-performance
- Easy to learn
- Powerful optimization

Solving with JuMP

```
model = Model(Ipopt.Optimizer)
@variable(model, x[1:2])
@objective(model, Min, (x[1] - 2)^2+(x[2]-3)^2)
@constraint(model, x[1] + x[2] == 1)
optimize!(model)
```

Examples

Aerospace

$$\begin{aligned}
\min_{x_{0:N},\, u_{0:N-1}} \quad & \left[\sum_{k=0}^{N-1} \|x_k - x_{ref}\|_Q^2 + \|u_k\|_R^2 \right] + \|x_N - x_{ref}\|_{Q_f}^2 \\
\text{s.t.} \quad & x_0 = x_{\text{init}} \\
& x_{k+1} = f(x_k, u_k), \quad k = 0, 1, \ldots, N-1 \\
& u_{min} \le u_k \le u_{max}, \quad k = 0, 1, \ldots, N-1 \\
& x_{min} \le x_k \le x_{max}, \quad k = 1, 2, \ldots, N
\end{aligned}$$

where:

- x_k is the state vector (e.g. position, velocity, attitude)
- u_k is the control input (e.g. thrust, torque)
- x_{ref} is the reference trajectory
- Q, R, Q_f are weighting matrices
- $f(x_k, u_k)$ represents aircraft dynamics
- State and control constraints ensure safety and feasibility

Rocket Landing

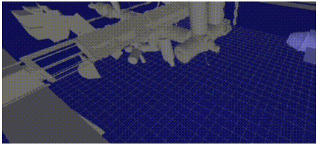


Figure 1: Spaceship

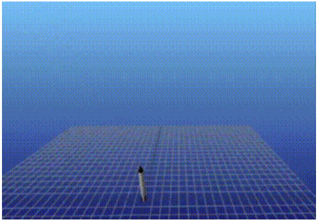


Figure 2: Rocket Landing

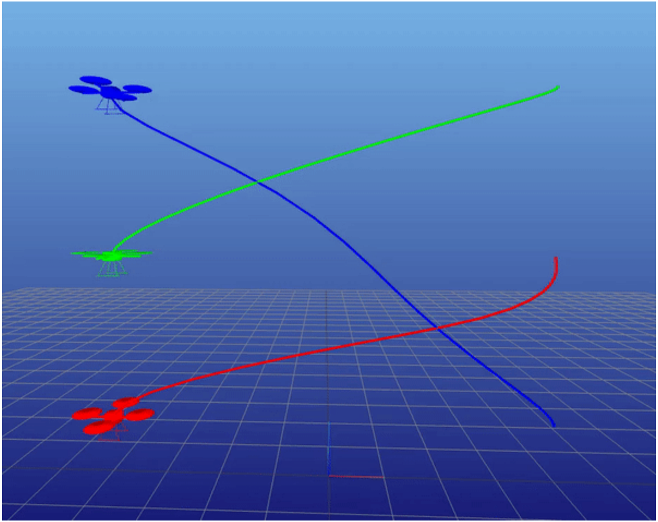
Quadrotors

$$\begin{aligned} \min_{x_{1:N},\, u_{1:N-1}} \quad & \left[\sum_{i=1}^{N-1} \ell(x_i, u_i) \right] + \ell_N(x_N) \\ \text{s.t.} \quad & x_1 = x_{\text{IC}} \\ & x_{k+1} = f(x_k, u_k), \quad \text{for } k = 1, 2, \ldots, N-1 \end{aligned}$$

where

- x_{IC} init
- $x_{k+1} = f(x_k, u_k)$ dynamc
- $\ell(x_i, u_i)$ cost
- $\ell_N(x_N)$ finalcost
- Safty Constraints

Quadrotors



Summary

Summary

- Optimal control is about finding the *best* way to control a dynamic system according to a defined objective (cost function).
- It involves understanding the system's dynamics, defining what "best" means (the cost function), and respecting constraints.
- It has broad applications across science, engineering, and economics.
- Solutions often require sophisticated tools like PMP, Dynamic Programming, or numerical optimization techniques.

Reference

[1] Zachary Manchester, “16-745: Optimal Control,” Carnegie Mellon University, 2023. Teaching Assistant: Kevin Tracy.

[2] Donald E. Kirk, “Optimal Control Theory: An Introduction,” Dover Publications, 2004.

[3] Zachary Manchester, Scott Kuindersma, “DIRTREL: Robust Trajectory Optimization with Ellipsoidal Disturbances and LQR Feedback,” Robotics: Science and Systems, 2018.

[4] Jorge Nocedal, Stephen J. Wright, “Numerical Optimization,” Springer Series in Operations Research, Second Edition, 2006.

Q&A

Thank you!

Your feedback will be highly appreciated!

Temporary page!

LaTeX was unable to guess the total number of pages correctl
As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many page to expect for this document.