Lab 4 Report

Robotics Integration Group Project I

Yuwei ZHAO (23020036096)
Group #31   2025-11-30

Abstract

This report investigates the mathematical foundations and practical application of polynomial trajectory optimization for UAVs. The primary objective is to formulate trajectory generation as a Quadratic Programming (QP) problem to minimize derivatives of position, such as velocity and snap. We begin by analytically deriving the cost and constraint matrixes for single-segment minimum velocity problems, verifying that the optimal solutions align with the Euler-Lagrange equation. The analysis is then extended to multi-segment minimum snap trajectories, identifying the necessary waypoint, continuity, and boundary constraints required for a unique solution. Finally, these theoretical frameworks are applied to a drone racing scenario, where optimal trajectories are generated to navigate a quadrotor through a sequence of gates.

See Resources on github.com/RamessesN/Robotics_MIT.

1 Introduction

Trajectory generation is a core component of quadrotor control, ensuring smooth navigation by minimizing specific state derivatives. This laboratory focuses on **Polynomial Trajectory Optimization**, specifically transforming the variational problem of minimizing an integral cost into a numerical Quadratic Program (QP).

The report is structured in three parts:

1. **Single-Segment Formulation**
   We analytically derive the cost matrix Q and constraint matrix A for a minimum velocity problem ($r = 1$). We verify that the QP solution aligns with the theoretical optimum derived from the Euler-Lagrange equation.
2. **Multi-Segment Extension**
   We extend the analysis to piece-wise polynomials over k segments. We derive the counting rules for waypoint, continuity, and boundary constraints to ensure a unique solution for high-order problems like Minimum Snap.
3. **Application**
   Finally, we utilize this framework to generate optimal trajectories for a drone racing scenario, navigating a quadrotor through a sequence of gates.

2 Procedure

2.1 Individual Work

2.1.1 Single-segment trajectory optimization

Consider the following minimum velocity (r = 1) single-segment trajectory optimization problem:

$$\min_{P(t)} \int_0^1 \left(P^{(1)}(t)\right)^2 dt \tag{1}$$

s.t.

$$P(0) = 0, \tag{2}$$

$$P(1) = 1, \tag{3}$$

with $P(t) \in \mathbb{R}[t]$, i.e., $P(t)$ is a polynomial function in t with real coefficients:

$$P(t) = p_N t^N + p_{N-1} t^{N-1} + \ldots + p_1 t + p_0 \tag{4}$$

Note that because of constraint (2) $P(0) = p_0 = 0$, and we can parametrize $P(t)$ without a scalar part p_0.

1. **Suppose we restrict $P(t) = p_1 t$ to be a polynomial of degree 1, what is the optimal solution of problem (1)? What is the value of the cost function at the optimal solution?**

   $\because P(t) = p_1 t$

   Let $t = 1 \therefore P(1) = p_1 \cdot 1 = p_1$
   $\because P(1) = 1 \therefore p_1 = 1$

   $\therefore$ optimal solution: $P(t) = t$.

   $\because P(t) = t \therefore P^{(1)} t = \frac{d}{dt} t = 1$

   $\therefore$ Cost $= \int_0^1 (1)^2 dt = 1$.

2. **Suppose now we allow $P(t)$ to have degree 2, i.e., $P(t) = p_2 t^2 + p_1 t$.**

   - **Write $\int_0^1 \left(P^{(1)}(t)\right)^2 dt$, the cost function of problem (1), as $p^T Q p$, where $p = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$ and $Q \in S^2$ is a symmetric 2×2 matrix.**

     $\because P(t) = p_2 t^2 + p_1 t \ \therefore P^{(1)}(t) = 2p_2 t + p_1$

     $\therefore$ Cost $= \int_0^1 (2p_2 t + p_1)^2 dt = p_1^2 + 2p_1 p_2 + \frac{4}{3} p_2^2$

In order to write into a 2×2 matrix as p^TQp, we have

$[p_1, p_2] \cdot \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = p_1^2 + 2p_1p_2 + \frac{4}{3}p_2^2$

$\therefore Q_{11} = 1, Q_{22} = \frac{4}{3}$

$\because Q_{12} = Q_{21} \ \therefore 2Q_{12} = 2$

$\therefore Q_{12} = Q_{21} = 1 \Rightarrow Q = \begin{bmatrix} 1 & 1 \\ 1 & \frac{4}{3} \end{bmatrix}$.

- **Write $P(1) = 1$, constraint (3), as $Ap = b$, where $A \in \mathbb{R}^{1\times 2}$ and $b \in \mathbb{R}$.**

$\because P(1) = 1$
$\therefore P(1) = p_2(1)^2 + P_1(1) = p_1 + p_2 = 1$

$\because Ap = b$ and $p = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \ \therefore A = [1\ 1], b = 1.$

- **Solve the Quadratic Program (QP):**

$$\min_p p^TQp \ s.t.\ Ap = b \tag{5}$$

You can solve it by hand, or you can solve it using numerical QP solvers (e.g., you can easily use the `quadprog` function in Matlab). What is the optimal solution you get for $P(t)$, and what is the value of the cost function at the optimal solution? Are you able to get a lower cost by allowing $P(t)$ to have degree 2?

$\because$ we have $\min_p p^TQp \Leftrightarrow \min\limits_{p_1, p_2}\left(p_1^2 + 2p_1p_2 + \frac{4}{3}p_2^2\right)$

and $Ap = b \Leftrightarrow p_1 + p_2 = 1$

Let $p_1 = 1 - p_2 \therefore$ Cost $= (1-p_2)^2 + 2(1-p_2)p_2 + \frac{4}{3}p_2^2 = 1 + \frac{1}{3}p_2^2$

In order to make `Cost` minimum $\Rightarrow \begin{cases} p_1 = 1 \\ p_2 = 0 \end{cases} \therefore \text{Cost}_{\text{minimal}} = 1.$

No, it remains the same value even though `P(t)` has degree 2.

3. **Now suppose we allow $P(t) = p_3t^3 + p_2t^2 + p_1t$:**

- **Let $p = [p_1 . p_2, p_3]^T$, write down $Q \in S^3, A \in \mathbb{R}^{1\times 3}, b \in \mathbb{R}$ for QP (5).**

$\because P(t) = p_3t^3 + p_2t^2 + p_1t$

$\therefore P_t^{(1)} = 3p_3t^2 + 2p_2t + p_1$

$\therefore \left[P_t^{(1)}\right]^2 = 9p_3^2t^4 + 4p_2^2t^2 + p_1^2 + \underbrace{12p_2p_3t^3}_{p_2p_3} + \underbrace{6p_1p_3t^1}_{p_1p_3} + \underbrace{4p_1p_2t}_{p_1p_2}$

$\therefore$ we have

$$\begin{cases} \text{item } p_3^2 : \int_0^1 9t^4 dt = \frac{9}{5} \\ \text{item } p_2^2 : \int_0^1 4t^2 dt = \frac{4}{3} \\ \text{item } p_1^2 : \int_0^1 1 dt = 1 \\ \text{item } p_2 p_3 : \int_0^1 12t^3 dt = 3 \\ \text{item } p_1 p_3 : \int_0^1 6t^2 dt = 2 \\ \text{item } p_1 p_2 : \int_0^1 4t dt = 2 \end{cases}$$

$$\therefore Q = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \frac{4}{3} & \frac{3}{2} \\ 1 & \frac{3}{2} & \frac{9}{5} \end{bmatrix}$$

$\because p_3(1)^3 + p_2(1)^2 + p_1(1) = 1 \Rightarrow 1 \cdot p_1 + 1 \cdot p_2 + 1 \cdot p_3 = 1$
$\therefore A = [1\ 1\ 1],\ b = 1.$

- **Solve the QP, what optimal solution do you get? Do this example agree with the result we learned from Euler-Lagrange equation in class?**

  From the above, we have the path that connects two points and minimizes the change in speed (energy) is always a straight line. That's regardless of the inclusion of higher-degree terms like t^2 or t^3, the optimization drives their coefficients to 0. The curve connecting the two points that minimizes the velocity cost is always a straight line. Consequently, the value of the cost function remains 1.

  Yes. By `Euler-Lagrange` equation, we have $\frac{\partial L}{\partial P} - \frac{d}{dt}\frac{\partial L}{\partial P'} = 0$

  Since $L = (P')^2 \ \therefore \frac{d}{dt}(2P') = 0 \Rightarrow P''(t) = 0$

  The condition $P''(t) = 0$ implies that the optimal function must be linear. The QP result is indeed a linear function, which confirms that the theroretical result derived from calculus of variations.

4. **Now suppose we are interested in adding one more constraint to problem (1):**

$$\min_{P(t)} \int_0^1 \left(P^{(1)}(t)\right)^2 dt,\ s.t. P(0) = 0,\ P(1) = 1,\ P^{(1)}(1) = -2 \tag{6}$$

Using the QP method above, find the optimal solution and optimal cost of problem (6) in the case of:

- $P(t) = p_2 t^2 + p_1 t$, **and**
- $P(t) = p_3 t^3 + p_2 t^2 + p_1 t.$

§ Case I. If $P(t) = p_2 t^2 + p_1 t$,

$$\begin{cases} \text{For } P(1)=1 : p_1 + p_2 = 1 \\ P^{(1)}(1) = -2 : p_1 + 2p_2 = -2 \end{cases}$$

$\therefore$ we have $\begin{cases} p_1=4 \\ p_2=-3 \end{cases}$

$\therefore \begin{cases} P(t)=-3t^2+4t \\ \text{Cost } =p_1^2+2p_1p_2+\frac{4}{3}p_2^2=4 \end{cases}$

§ Case II. If $P(t) = p_3t^3 + p_2t^2 + p_1t$,

$\begin{cases} \text{For } P(1)=1: p_1+p_2+p_3=1 \\ P^{(1)}(1)=-2: p_1+2p_2+3p_3=-2 \end{cases}$

$\therefore \begin{cases} p_1=4+p_3 \\ p_2=-3-2p_3 \end{cases}$

From (3), we have: $Q = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \frac{4}{3} & \frac{3}{2} \\ 1 & \frac{3}{2} & \frac{9}{5} \end{bmatrix}$

$$p = \begin{bmatrix} 4+p_3 \\ -3-2p_3 \\ p_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 4 \\ -3 \\ 0 \end{bmatrix}}_{p_{\text{base}}} + p_3 \underbrace{\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}}_{d}$$

For $\text{Cost}(p_3) = Ap_3^2 + Bp_3 + C$, we have $\begin{cases} A=d^TQd \\ B=2p_{\text{base}}^TQd \end{cases}$

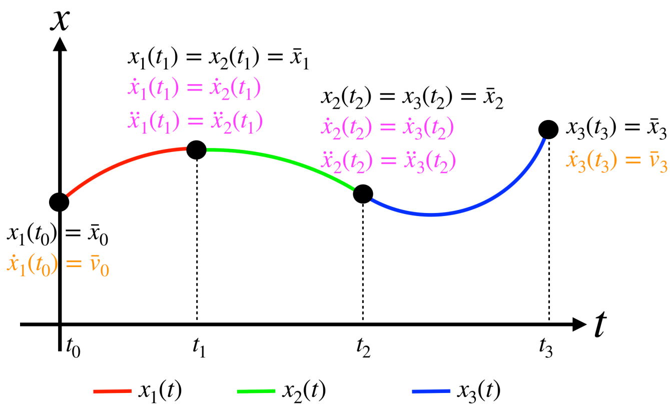
$\therefore Qd = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \frac{4}{3} & \frac{3}{2} \\ 1 & \frac{3}{2} & \frac{9}{5} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{6} \\ -\frac{1}{5} \end{bmatrix}$

$\therefore A = \frac{2}{15},\ B = 1 \ \therefore \text{Cost} = \frac{2}{15}p_3^2 + p_3 + 4$

In order to make `Cost` minimum $\Rightarrow p_3 = -\frac{15}{4} \Rightarrow \text{Cost}_{\text{minimal}} = \frac{17}{8}$.

2.1.2 Multi-segment trajectory optimization

1. **Assume our goal is to compute the minimum snap trajectory ($r = 4$) over k segments. How many and which type of constraints (at the intermediate points and at the start and end of the trajectory) do we need in order to solve this problem? Specify the number of waypoint constraints, free derivative constraints and fixed derivative constraints.**



$$\text{Cost} = \int \left(x^{(4)}(t)\right)^2 dt \Rightarrow x^{(2r)}(t) = 0 \Rightarrow x^{(8)}(t) = 0$$

Step 1: Determine the number of Unknowns

Given the cost function $\text{Cost} = \int \left(x^{(4)}(t)\right)^2 dt$, the Euler-Lagrange equation yields the necessary condition:

$$x^{(2r)}(t) = 0 \overset{r=4}{\Rightarrow} x^{(8)}(t) = 0$$

Integrating this equation **8 times**, we obtain a polynomial of degree $2r - 1 = 7$:

$$P(t) = p_7 t^7 + p_6 t^6 + ... + p_1 t + p_0$$

$\because$ Each segment has $N + 1 = 8$ unknown coefficients and there are k segments.
$\therefore$ Total Unknowns = $8k$, which means we need $8k$ constraints to solve for a unique solution.

(1) For *Waypoint Constraints*:
For each segment i, the position at start t_{i-1} and end t_i is fixed.
$\therefore$ 2 constraints $\times$ k segments = $2k$ constraints.

(2) For *Free Derivative Constraints*:
At the $(k-1)$ intermediate waypoints, the trajectory must be smooth.
Continuity is required for derivatives up to $2r - 2 = 6$ (i.e., 1^{st} to 6^{th} derivatives).
$\therefore$ 6 constraints $\times$ $(k-1)$ points = $6(k-1)$ constraints.

(3) For *Fixed Derivative Constraints*:
At the start t_0 and end t_k of the entire trajectory, we fix derivatives up to $r - 1 = 3$ (Velocity, Acc, Jerk).
$\therefore$ 3 (start) $+3$ (end) = 6 constraints.

> Proof:
$2k + 6(k-1) + 6 = 8k$, which confirms that total number of constraints is $8k$.

2. **Can you extend the previous question to the case in which the cost functional minimizes the r-th derivative and we have k segments?**

From the method above, we have $\text{Total number of constraints} = 2rk$.
Specifically,

(1) For *Waypoint Constraints*:
$2k$ (Start and End positions for each segment).

(2) For *Free Derivative Constraints*:
$(k-1) \cdot (2r-2)$ (Continuity of 1^{st} to $(2r-2)^{\text{th}}$ derivatives at intermediate points).

(3) For *Fixed Derivative Constraints*:
$2(r-1)$ (Fixing 1^{st} to $(r-1)^{\text{th}}$ derivatives at t_0 and t_k).

> **Proof:**
$2k + (k-1)(2r-2) + 2(r-1) = 2rk$, which confirms that total number of constraints is $2rk$.

2.2 Team Work

2.2.1 Drone Racing

3 Reflection and Analysis

4 Conclusion

5 Source Code

-
-