Project Report

Xiaochen Li

December 12, 2023

1 Abstract

This paper will focus on the load-carrying capability of a quadrotor. The optimal control problem is formed to find the optimal trajectory as well as the maximum cable-suspend load for a quadrotor to carry from one point to another point in free space. Pontryagin's Minimium Principle(PMP) was used to derive the optimal control input u, and transfer the problem into a two-point boundary value problem (TPBVP), which was solved by successive approximation method. The results was verified by the simulation plot.

2 Introduction

Quadrotor, also known as unmanned aerial vehicles (UVAs), possess distinctive capabilities such as hovering, vertical takeoff, flight, and precise vertical landing in confined spaces. These attributes make them well-suited for a variety of military and civilian operations, including photography, inspections, research platforms, relief efforts, operations in high-risk areas, and cargo transportation, among other applications. Cargo transportation capability has been specifically highlighted recently as it is crucial to deliver key supplies like first-aid kits during natural disasters like earthquakes. Trajectory planning for such a task needs to be well defined for a safe and efficient delivery. However, the dynamics for a cable-suspend load on a quadrotor were inherently complex, as there are 8 state variables that descirbes the equation of motion. Therefore, optimal control was used to solve this complicated problem.

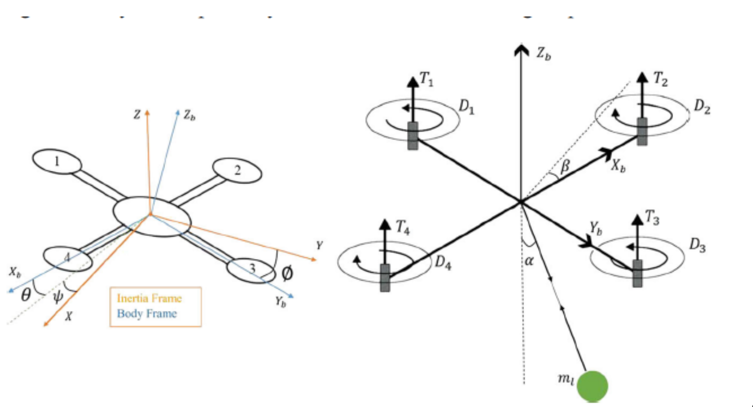

Figure 1: Quadrotor States

The equation of motion(EOM) of the system was first derived by Euler-Langrange equation. The cost functional was then defined as a trade-off between controlling states and the control effort being applied. The control input u was defined as the thrust force provided by the propeller. The optimization objective is to find the minimized cost such that the quadrotor could fly from one point to another, with the cable-suspended load. The cost functional was subjected the the dynamics of the quadrotor, as well as the saturation of propeller thrust, which will be defined later. The problem was then formed to a boundary value problem (BVP) by applying Pontryagin's Minimum Principle (PMP). Hamiltonian was calculated and coupled non-linear ODE's of all states and co-states were solved together. Matlab was used as the solver to run the minimization, with boundary conditions being defined. Successive approximation was used to solve this problem due to the complicated dynamics as well as the large number of states and co-states. After the optimal trajectory was found, different payload mass was tested from 0 to a certain value.

3 Problem Formulation

Quadrotors have total of 4 motors which drive the propeller to provide thrust force. Moving forward and backward, as well as rotation were achieved by changing the angle of the rotors such that the thrust force could be in any direction. Figure 1 shows a brief diagram of the quadrotor states, where x, y, z represented the position, and the ψ, θ, ϕ represented the rotational angles about x, y, z, which were also known as yaw pitch and roll. Assuming constant cable length r, angle α and β describe the rotation of the mass load relative to the quadrotor. Therefore, our state variables could be written as:

$$q = [x\ y\ z\ \psi\ \theta\ \phi\ \alpha\ \beta]^T \in R^{8\times 1} \tag{1}$$

A couple assumptions need to be made before deriving the dyanmics model:

- Cable mass was ignored
- The load could be considered as a point mass
- The cable is inelastic. In other words, the length of the cable will not change.
- Thrust and drag constant are the same for all propellers.
- Mass distribution for the quadrotor are the same on x-y plane, i.e. $I_{xx} = I_{yy}$
- Rigit body.

The non-linear dynamics model for quadrotor with cable-suspended load could be generally represented as follows :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = U \tag{2}$$

where:

- $M(q) \in R^{8\times 8}$ is the mass matrix
- $C(q, \dot{q}) \in R^{8\times 8}$ is the Coriolis and centrifugal matrix
- $G(q) \in R^{8\times 1}$ is the gravity matrix
- $U \in R^{8\times 1}$ is the generalized force in which $U = b \times u$

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \tag{3}$$

Therefore, we could dervice our dynamic equations as:

$$\dot{X} = F(X, U) \tag{4}$$

$$\dot{X}_1 = X_2 \tag{5}$$

$$\dot{X}_2 = -M^{-1}(C(X_1, X_2)X_2 + G(X_1)) + M^{-1}U \tag{6}$$

The main goal of our optimization is to find the optimal path that move the quadrotor from one initial point to the desired final destination, with position and orientation being specified. Hence, the cost functional needs to focus on both controlling the states, and the control effort being applied. The cost functional J could be written as follows:

$$J\left(U, m_l\right)=\int_{t_0}^{t_f} L\left(X, U, m_l\right) d t=\frac{1}{2}\left\|X_1\right\|_{W_1}^2+\frac{1}{2}\left\|X_2\right\|_{W_2}^2+\frac{1}{2}\|U\|_R^2 \tag{7}$$

Where the norm $\frac{1}{2}\left\|X_1\right\|_{W_1}^2=X^T K X$. Here, the matrix W_1, W_2 and R are symmetric, positive-definite matrix that is used to adjust the weight for each factor. The higher the value in the matrix, the higher the relative importance. Then the whole problem could be written as an optimization problem as follows:

$$\begin{array}{ll}
\min & J\left(U, m_l\right)=\int_{t_0}^{t_f} L\left(X, U, m_l\right) d t \\
\\
\text{s.t.} & \dot{X}=F(X, U) \quad x \in \mathbb{R}^8 ; u \in \mathbb{R}^8 \\
& U^{-} \leq u \leq U^{+}
\end{array} \tag{8}$$

with boundary conditions $X(t_0)=X_0$ and $X(t_f)=X_f$, which describes the initial and terminal position and orientation that was trying to achieve. The control input u was constrained to be smaller than the maximum allowable value of the rotor. Pontryagin's Minimum Principle was then used to form the essential condition for optimality, and coupled non-linear ODE's between states and co-states were set up for later use. The Hamiltonian was derived as follows:

$$H\left(X, U, P, m_l, t\right)=\mathrm{L}\left(X, U, m_l\right)+P^T(t) F\left(X, U, m_l\right) \tag{9}$$

and the essential conditions for optimality were derived as follows: The optiaml trajectory and control will be derived by solving this coupled non-linear ODE's

$$\begin{aligned}
& \dot{X}^*(t)=\frac{\partial H}{\partial P}\left(X^*(t), U^*(t), P^*(t), t,\right) \\
& \dot{P}^*(t)=-\frac{\partial H}{\partial X}\left(X^*(t), U^*(t), P^*(t), t,\right) \\
& H\left(X^*(t), U^*(t), P^*(t), t\right) \leq H\left(X^*(t), \bar{U}(t), P^*(t), t\right)
\end{aligned} \tag{10}$$

Where symbol $*$ notes to the extremals of $X(t)$, $U(t)$ and $P(t)$ The optimal control problem was now being transformed to a two-point boundary value problem(TPBVP), which is relatively hard to solve. Fortunately, the matlab function BVP4C was able to solve the boundary value problem numerically, which will be discussed in detail later.

Parameter	Value
m(kg)	0.56
1(m)	0.242
r(m)	1
I_l ($\mathrm{kg} \cdot \mathrm{m}^2$)	$m_l l^2$
$I_x=I_y$ ($\mathrm{kg} \cdot \mathrm{m}^2$)	6.178e − 3
I_z ($\mathrm{kg} \cdot \mathrm{m}^2$)	2.1e − 3
$u_1(N)$	$-3N \leq u_1 \leq 12$
$u_2(N)$	$-6N \leq u_2 \leq 6$
$u_3(N)$	$-6N \leq u_3 \leq 6$
$u_4(N)$	$-6N \leq u_4 \leq 6$

(11)

4 Method

4.1 General Problem-Solving Procedure

The whole system consists of 16 states (counting both q and $\dot{q}$, therefore there are total of 16 co-states: P_1 through P_{16}. The whole problem-solving procedure was as follows:

1. First, declare all symbolic variables using *syms* in Matlab, including all states, co-states and 4 control inputs from u_1 to u_4.
2. Plug in those variables and constants shown in table 11 into the dynamic equation shown in 4
3. Plug in variables to form the cost functional J, and derive the Hamiltonian
4. Find the derivative of the co-states by using Matlab "diff" function, which was later used for solving the couple non-linear ODEs.
5. Derive the optimal u by differentiating calculating $\frac{\partial H}{\partial U}$ and set it to 0. Saturations were set to avoid u larger than the boundary.
6. Copying the dynamic formulation to form a function ready to be put into BVP4C to solve. Both dynamics of the quadrotor and co-states were listed.
7. Set up boundary conditions for both states and co-states.
8. Put everything into the Matlab solver and derive the solution.

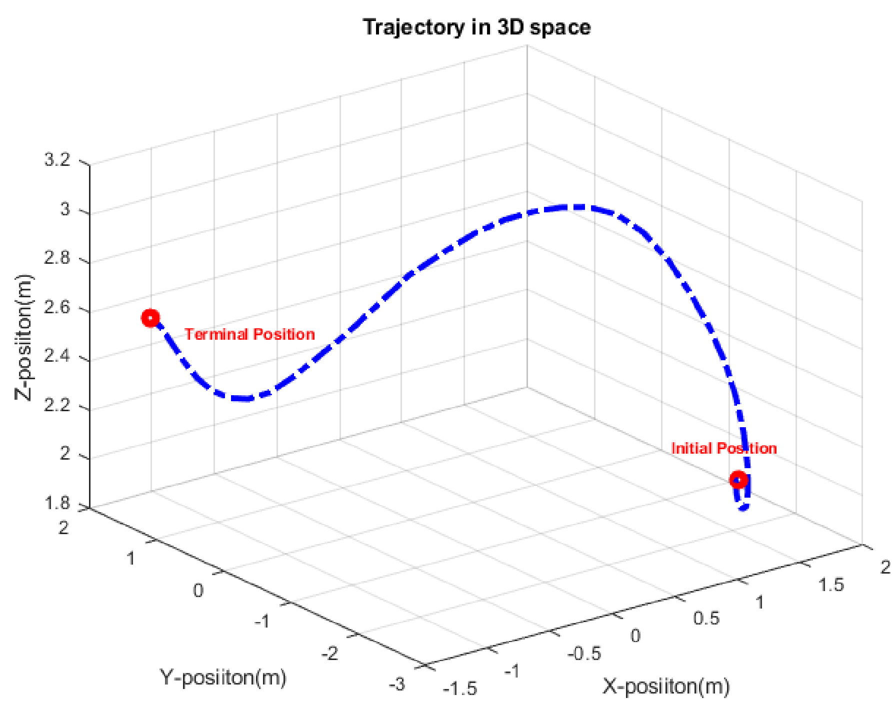

Figure 2: Optimal Trajectory in 3D space

4.2 Setting up TPBVP

Since our objective of this optimal control problem is to find a control input $u(t)$ such that it minimizes the cost functional J, and also drives the system from its initial position to the terminal position. Therefore for

this specific system, the initial condition and terminal condition for all states are fixed. For example, we would like the quadrotor initially from the point [1.5 -2 2] to [-1 2 2.5], with all other states from 0 to 0. In other words, in order for safe and efficient transportation, we would like the quadrotor to have 0 in rotation and have both initial and terminal velocity to be 0. Same for the angles of the load.
Considering we have total of 32 states(states + costates), it will require the solver to have 32 boundary conditions to solve. The first 16 conditions were given as the initial conditions for all states. The x y z position was selected to be [1.5 -2.5 2], with the rest of the states to be 0. The terminal states for all costates were selected to be 0. However, the optimal u does not ensure the correctness of the terminal states, i.e. such u might not be able to drive the system to the desired position, as well as desired orientation and velocity. Therefore, a terminal cost that looks like the following needs to be added into the cost functional J:

$$h\left(X\left(t_f\right), t_f\right)=\frac{1}{2}\left\|X_1\left(t_f\right)-X_{1f}\right\|_{W_p}^2+\frac{1}{2}\left\|X_2\left(t_f\right)-X_{2f}\right\|_{W_v}^2 \tag{12}$$

4.3 Successive Approximation

Although easy to implement, using bvp4c usually returns failed to solve due to the singularity of the Jacobian matrix. Such situation is highly likely to be caused by the incorrect initial guess of the solution. The correctness and robustness of the bvp4c solver highly depend on the initial guess. Therefore, successive approximation is then implemented. The general procedure could be written as follows:

1. Divide interval $[t_0, t_f]$ into N equal subintervals and assume a piecewise-constant control $u(t) = u\left(t_k\right), \quad t \in \left[t_k, t_{k+1}\right] \quad k=0,1,\cdots,N-1$
2. Applying the assumed control u to integrate the state equations from t_0 to t_f with initial conditions $\mathrm{x}\left(t_0\right)=\mathrm{x}_0$ and store the state trajectory x.
3. Applying u and x to integrate costate equations backward, i.e., from $[t_f, t_0]$. The "initial value" $\mathbf{p}\left(t_f\right)$ can be obtained by:
$$\dot{\mathbf{P}}\left(t\right)=\frac{\partial H(t)}{\partial \mathbf{x}(\mathbf{t})}$$
Evaluate $\partial H^{(i)}(t)/\partial u, t \in [t_0, t_f]$ and store this vector. Here, we will use 0 as the "initial condition" (actually the terminal condition) for all costates
4. If
$$\left\|\frac{\partial H}{\partial u}\right\| \leq \epsilon$$
then stop the iterative procedure. Here ϵ is a constant for measuring the precision of the loop.
If not satisfied, adjust the piecewise-constant control function by:
$$u^{(i+1)}\left(t_k\right)=u^{(i)}\left(t_k\right)-\tau \frac{\partial H^{(i)}}{\partial u}\left(t_k\right), \quad k=0,1,\cdots,N-1$$
Replace $u^{(i)}$ by $u^{(i+1)}$ and return to step 2 . Here, τ is the step size. Same with using bvp4c, to ensure the terminal value for our states is exactly what we want, a terminal cost shown in equation 12 was also added to the cost functional

The psudo-code for implementing successive approximation was shown in algorithms 1 The results will be shown in the next section

5 Results and Conclusion

Both figure 2 and 3 shows the optimal trajectory for the quadrotor goes from [1.5 -2 2] to [-1 2 2.5]. It could be observed that the error at the terminal states is very minor, which means a very good precision for your optimal controller. The payload mass was chosen as 0.1 kg.

Algorithm 1: Successive Approximation for Optimal Control

Initialize: t_0, t_f N, initx, initp

tspan = linsapce(t_0, t_f, N)

Initialize: u = ones(4,N)

while $\left\| \frac{\partial H^{(i)}}{\partial u} \right\| \leq \epsilon$ **do**

Forward Integration of State Equations: Integrate $\dot{x} = f(x, u)$ from $x(0)$ to obtain $x(t)$.

Backward Integration of Costate Equations: Define the Hamiltonian $H = L(x, u) + P^T f(x, u)$. Integrate $\dot{P} = -\frac{\partial H}{\partial x}$ backward in time.

Update Control: Update $u(t) = \arg\min_u H(x(t), u, P(t))$.

Convergence Check: Check for convergence.

Terminate: Output the converged control $u(t)$.

To find the maxmium payload the quadrotor is able to carry, a loop was written and the boundary value problem was kept solving with increasing payload until one of the control u reached the limit. The load starts at 0.1 kg and kept raising with increments of 0.02 kg. The ultimate load before u_1 reaches its upper limit is 0.28kg, which is slightly different from the paper. Which might be due to some numerical error.

In conclusion, our main objective is to find an optimal trajectory for a quadrotor to carry a cable-suspend load safely (with velocity being 0 at t_0 and t_f) from one point to another point in free space. Dynamic of this combined system was first derived by the Euler-Langrange formulation. The problem was then formulated using Pontryigin's Minimum Principle such that the optimization could be transferred to finding solutions with coupled non-linear ODE's between states and costates. The optimal control input could be obtained by $u(t) = \arg\min_u H(x(t), u, P(t))$. Successive Approximation method was then applied to find the optimal solution iterativly.

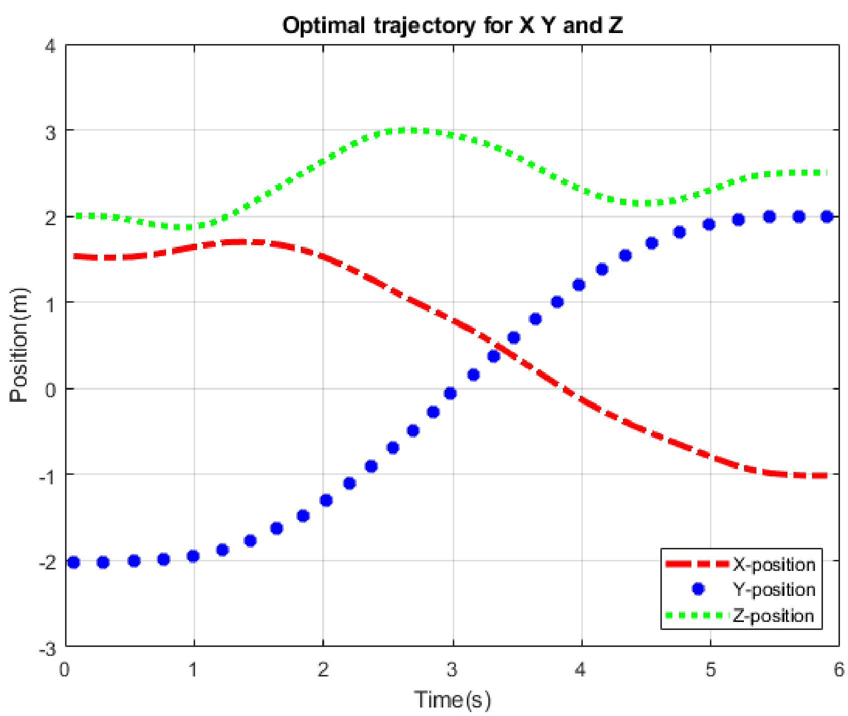

Figure 3: X Y and Z positions for optimal trajectory

References

[1] Hashemi, D., Heidari, H. (2020). Trajectory Planning of Quadrotor UAV with Maximum Payload and Minimum Oscillation of Suspended Load Using Optimal Control. Journal of Intelligent Robotic Systems., 100(3–4), 1369–1381. https://doi.org/10.1007/s10846-020-01166-4