Trajectory Optimization of a Quadrotor with Cable Suspended Load

Brian Bittner, Quan Nguyen

Abstract—In this paper trajectory optimization is applied toward producing aggressive, precise trajectories for a quadrotor with suspended load. Using differential flatness, the system dynamics can be derived from a smooth, continuous load trajectory. This allows for simple construction of an optimization problem to generate a thrust-optimal load trajectory for waypoint navigation. We present such a method for determining trajectories in set of various situations. After successfully solving the waypoint navigation problem, we will test trajectory generation in the midst of obstacles. Additionally, we test a controller to ensure that the system is robust against initial condition error and pose estimation noise.

I. INTRODUCTION

Consider highly trained helicopter pilots, who pick up tree trunks with suspended cables and swing them into truck beds. Such a maneuver requires execution of an aggressive, smooth, precise trajectory. We look to mimic such behavior in a scaled down system. We simulate a quadrotor that weighs 0.5kg. A 0.09kg load swings 20cm below the base of the quadrotor. Recent work has been done on the window problem, which involves navigation through a vertical space smaller than the length of the cable. We will work towards obstacle laden environments such as this, which require highly aggressive, precise movement.

The differential flatness of our system is vital to our problem formulation. It allows us to generate continuous load position and yaw trajectory functions which ensure differential feasibility for our system. By checking constraints on thrust and cable tension we can assert the trajectory is dynamically feasible. The only information required for these assertions are the load position and yaw trajectories.

The layout of our paper is as follows: In Section 2 we cover mechanics of the quadrotor with suspended load. In Section 3 we formulate our optimization technique for generating load trajectories. In Section 4 the controller is presented. Section 5 contains our results, which will include trajectory generation for waypoint navigation and obstacle avoidance. This section will also evaluate controller performance on the trajectories produced by the optimizer. Section 6 concludes the paper.

II. SYSTEM DYNAMICS

The dynamics and the controller of our project are based on [2]. The dynamics of the system should be hybrid due to the switching behavior between the two dynamics model: Non-zero cable tension and zero cable tension. However, for simplification, in our project, we consider only the non-zero cable tension case and guarantee it by constraints in the optimization.

In order to describe the dynamic of the system in Fig 1, we define

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_Q \in \mathbb{R}^3$</td>
<td>Position of the center of mass of the quadrotor</td>
</tr>
<tr>
<td>$v_Q \in \mathbb{R}^3$</td>
<td>Velocity of the center of mass of the quadrotor</td>
</tr>
<tr>
<td>$R \in \text{SO}(3)$</td>
<td>Rotation matrix of the quadrotor from body-fixed frame to the inertial frame</td>
</tr>
<tr>
<td>$\Omega \in \mathbb{R}^3$</td>
<td>Angular velocity of the quadrotor in the body-fixed frame</td>
</tr>
<tr>
<td>$x_L \in \mathbb{R}^3$</td>
<td>Position of the suspended load</td>
</tr>
<tr>
<td>$v_T \in \mathbb{R}^3$</td>
<td>Velocity of the suspended load</td>
</tr>
<tr>
<td>$l \in \mathbb{R}$</td>
<td>Length of the suspension cable</td>
</tr>
<tr>
<td>$T \in \mathbb{R}$</td>
<td>Tension in the cable</td>
</tr>
</tbody>
</table>

The system dynamics of the quadrotor with cable-
suspended load is obtained as

\[ \dot{x}_L = v_L, \]
\[ (m_Q + m_L)(\dot{v}_L + g e_3) = (q \cdot f Re_3 - m_Q l \dot{q}) q, \]
\[ \dot{q} = \omega \times q, \]
\[ m_Q l \dot{\omega} = -q \times f Re_3, \]
\[ \hat{R} = R \hat{\Omega}, \]
\[ J_Q \dot{\Omega} + \Omega \times J_Q \Omega = M. \]

### III. Trajectory Optimization Method

#### A. Differential Flatness

In our project, we used the method of differential flatness to construct the optimization problem. To be more specific, since the system is differentially flat [2], we are able to construct the trajectory of the whole system \((x_Q, R, x_L)\) and the inputs as well \((f, M)\) based on the trajectory of the suspended load \(x_L(t)\) and its higher order derivatives \(\dot{x}_L(t), \ddot{x}_L(t), x_L^{(3)}(t), \ldots\) (with this system we need up to 6th order derivative). Yaw affects which combination of the four fans produces the thrust. In other words, yaw can affect the orientation of the moment on the quadrotor with respect to the body frame, but not its magnitude. Because this does not affect the dynamics in a way that is meaningful for our analysis, we set it to always be 0 degrees. From this property, we can easily construct the optimization by parameterizing the trajectory of the suspended load. In our project, we tried the simple case of using polynomial

\[
x_Q = \sum_{i=1}^{m} p_i^x t^i
\]
\[
y_Q = \sum_{i=1}^{m} p_i^y t^i
\]
\[
z_Q = \sum_{i=1}^{m} p_i^z t^i
\]

The role of the optimization is now just finding a small set of parameter with \((3 \times m)\) elements

\[
P = \begin{bmatrix} p_1^x & p_2^x & \ldots & p_m^x \\ p_1^y & p_2^y & \ldots & p_m^y \\ p_1^z & p_2^z & \ldots & p_m^z \end{bmatrix}
\]

As a result, we now have

\[
\begin{bmatrix} x_Q \\ v_Q \\ R \\ \Omega \\ x_L \\ v_L \\ f \\ M \\ T \end{bmatrix} = f(P, t)
\]

#### B. Optimization Setup

Parameter set \(P\) fully describes our system dynamics. We formulate our optimization as finding the set of \(m^{th}\) order derivative polynomial coefficients that produce a minimum thrust load trajectory.

For all scenarios in which load trajectories were optimized, it is required that the load and quadrotor start from rest and come to a stop at a set of waypoints.

1) Cost Function: The criterion for our optimization function is to minimize the total thrust exerted by the four quadrotor fans. This scoring method is simplistic, but provides a proper guide for load trajectory construction. A minimum-thrust criterion will push our solution toward having desirable trajectory characteristics, such as smoothness and short path length.

To calculate the score we sample the thrust trajectory \(n\) times across the duration. We sum the the square of sampled thrusts.

\[
J = \sum_{i=1}^{n} f(t_i)^2
\]

A less-smooth trajectory could be produced by a criterion based on maintaining distance from an obstacle or minimizing duration. Any tendency in the cost function to create a high acceleration movement will directly increase the required thrust, as indicated by the differential flatness dynamics.

2) Constraints: Constraints are first formulated to maintain that our dynamic model is valid for the duration of the flight. Because we do not apply the hybrid-model for a tense and slack cable, we must ensure that the cable is tense at all times. To ensure this we sample the tension trajectory \((T)\) \(n\) times, imposing an inequality at each point that the tension remains positive.

\[
T > 0
\]

Additionally, we need to ensure that the thrust produced by the quadrotor is also positive, since a negative thrust is not physically achievable. Similarly we sample the thrust trajectory \((f)\) \(n\) times.

\[
f >= 0
\]

The maximum thrust produced by the quadrotor must be set with respect to mechanical configuration and power limitations. Across the community it is reasonable to expect a maximum thrust of 2.5 times the quadrotor weight. We sample the thrust trajectory \((f)\) \(n\) times to impose an inequality constraint such that the thrust does not exceed 12.5 Newtons (0.5kg quadrotor.)

\[
f <= 12
\]
The quadrotor and load have specified start and end waypoints. They must also obtain zero velocity at these points. The following equality constraints were constructed.

\[
\begin{align*}
X_Q(t_n) - X_{Q_{\text{final}}} &= 0 \quad (16) \\
X_Q(t_0) - X_{Q_{\text{init}}} &= 0 \quad (17) \\
V_Q(t_n) &= 0 \quad (18) \\
V_Q(t_0) &= 0 \quad (19) \\
X_L(t_0) - X_{L_{\text{init}}} &= 0 \quad (20) \\
X_L(t_n) - X_{L_{\text{final}}} &= 0 \quad (21) \\
V_L(t_0) &= 0 \quad (22) \\
V_L(t_n) &= 0 \quad (23)
\end{align*}
\]

C. Avoiding Obstacle

Obstacle Avoidance is the most computationally intensive portion of our optimization computation. Parametric curves define the spatial trajectory of the quadrotor and load. Both curves are sampled to ensure that it does not intersect some set of spatial constraints defining the obstacle.

The most common type of obstacle we generate is a sphere. Using \(X_Q\) and \(X_L\) we can check to ensure that the quadrotor and load do not intersect the spherical obstacle, but this is not good enough. It needs to be ensured that no part of the tether comes into contact with the obstacle, as contact would greatly alter the dynamics. Rather than sampling a large quantity of points along the tether, we construct a way to sample only 4 intermediate points.

We place 6 spheres starting at the quadrotor center of mass, along the cable, and ending at the load center of mass. The radial length of each sphere is set such that the spheres are all perfectly adjacent to each other. An inequality constraint ensures that the spheres do not intersect any obstacles. This is also sampled \(n\) times along the six trajectories.

\[|X - \text{obs}| - \text{rad}_X - \text{rad}_\text{obs} > 0 \quad (24)\]

This formulation serves two purposes. We are able to test a much smaller number of points to guarantee that our system stays clear of all obstacles. This also applies a radial collision buffer to the quadrotor and load.

IV. CONTROLLER

In order to improve the reliability of the method, we apply feedback control to tracking the optimal trajectory. Due to the complexity, nonlinearity and under-actuation of the system, a small error from model and initial conditions will cause a large tracking error from the desired trajectory during real world experimentation.

We used PD-based controller presented in [2] with the control diagram illustrating in Fig. 2. Another type of controller that could be used for this problem is geometric control [1].

We then test the robustness of our approach by presenting strong perturbation on initial condition and adding noises to the system.

Fig. 2: Control Diagram

V. RESULTS

In this Section, we will present simulation result with both trajectory optimization and feedback control implementation for different cases as follows.

<table>
<thead>
<tr>
<th>Case</th>
<th>Optimization with fixed duration</th>
<th>Optimization with optimizing duration with carrying load from (x_{L_i} = [0; 0; 0]) to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 2</td>
<td>(x_{L_f} = [10; 0; 0])</td>
<td>Case 21: (x_{L_f} = [10; 0; 0])</td>
</tr>
<tr>
<td>Case 2</td>
<td>(x_{L_f} = [3; 2; 0])</td>
<td>Case 22: (x_{L_f} = [3; 2; 0])</td>
</tr>
<tr>
<td>Case 2</td>
<td>(x_{L_f} = [5; -2; 0])</td>
<td>Case 23: (x_{L_f} = [5; -2; 0])</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 3</th>
<th>Optimization with avoiding sphere obstacle with different radius (r) (the cable length is reduced from (L = 1[m]) to (L = 0.2[m]) for this case.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 3</td>
<td>(r = 0[m])</td>
</tr>
<tr>
<td>Case 3</td>
<td>(r = 0.75[m])</td>
</tr>
<tr>
<td>Case 3</td>
<td>(r = 1.75[m])</td>
</tr>
</tbody>
</table>

The quadrotor model is used to test in our project is a Hummingbird quadrotor (see Fig.1). The model parameters are given as follows.

- **Mass of the quadrotor**: \(m_Q = 0.5\) [kg]
- **Inertia matrix of the quadrotor with respect to the body-fixed frame**: 
  \[
  J_Q = \begin{bmatrix}
  2.32e^{-3} & 0 & 0 \\
  0 & 2.32e^{-3} & 0 \\
  0 & 0 & 4e^{-3}
  \end{bmatrix}
  \]  [kg.m²]
- **Mass of the suspended load**: \(m_L = 0.09\) [kg]
- **Length of the suspension cable for the problem of NO obstacle**: \(L = 1[m]\)
- **Length of the suspension cable for the problem of avoiding obstacle**: \(L = 0.2[m]\)

In the first phase of carrying the suspended load from "Start" to "Goal" we tried two cases. In Case 1, with fixed duration, the solution is quite sensitive with the initial guess and we need to design carefully such a set to be able to handle different desired "Goal". Fig. 3 is one successful exam. The most difficult challenge that limits significantly the feasibility of the problem is that we need to guarantee the whole system (both the quadrotor and the suspended load need to be stopped at the end of the process). As a result, a nice property of motion can be observed from different simulation cases (Fig. 4, 6) that the quadrotor need to lean backward a bit when it closed to the "Goal" so that it can
stop the suspended load at right position and at the same time the quadrotor can stop above the load.

In order to improve the feasibility, we consider Case 2 of optimizing the timing also. This approach can easily handle nearly all sets of “Goals” with just simple initial guess of all zeros. Fig. 5 showed three cases.

We need to notice that from Fig. 3, 5, our trajectory always guarantee desired constraints such that positive tension \( T > 0 \), input saturation, go from “Start” to “Goal” and stop with zero velocity at the end.

Based on the same approach, we can also be able to handle large scale of obstacle (up to a big sphere of 1.5[m] at the middle of the ”Start” and “Goal”) (see Fig. 8). In this case, because the cable length of \( L = 1[m] \) seems to be too aggressive and it was difficult to give us feasible solution, we decided to reduce the cable length to \( L = 0.2[m] \). Physically, this is already a very challenging problem.

About control implementation, Fig. 4, 6, 9 showed results with perfect initial condition and strong perturbation on it for three optimization cases Case 1 (fixed duration), Case 2(optimizing duration), Case 3(avoiding obstacle). In Fig.7, we added noise to the system. Tracking performance of the closed-loop system is very good in these cases, proving the robustness of our method.
Fig. 5: Case 2: Optimization with optimizing duration for carrying the load to different goals.

Fig. 6: Control Implementation for Case 2 (Optimization with optimizing duration)

Fig. 7: Control Implementation with noise

\( \hat{x} = x + 0.05 \sin(5t) \times x + 0.02 \sin(20t) \times \text{Ones}(n, 1) \).
Fig. 8: Case 3: Optimization with avoiding obstacle

Fig. 9: Control Implementation of Case 3 (Optimization with avoiding obstacle). Because the cable length of $L = 1[m]$ is too aggressive in this case, we reduce the cable length to $L = 0.2[m]$. 
VI. FUTURE WORK

Looking forward we look to use differential flatness and maximum thrust knowledge to generate more aggressive trajectories for tougher sets of obstacles, such as the window problem.

Once able to produce such aggressive trajectories, we look forward to testing on a hummingbird.

VII. CONCLUSION

We present a set of high performance trajectories generated for various maneuvers. In cases of waypoint navigation and obstacle avoidance, we consistently generate trajectories that obey all system constraints and minimize thrust. Differential flatness of our system allows us to formulate load trajectories that guarantee differential feasibility, greatly reducing the complexity of our optimization problem formulation.

Our controller is tested on the set of planned trajectories, encountering initial state error, perturbations, and pose estimation noise. The performance across these cases indicates that the controller and trajectories generated are robust against an impressive range of disturbance magnitudes and types.

REFERENCES
