

Article

Optimal Aggressive Constrained Trajectory Synthesis and Control for Multi-Copters

Tsung-Liang Liu [†] and Kamesh Subbarao ^{*,†} 

Department of Mechanical and Aerospace Engineering, The University of Texas at Arlington, 500 W. First St, Box 19018, 211 Woolf Hall, Arlington, TX 76019, USA; tsungliang.liu@mavs.uta.edu

* Correspondence: subbarao@uta.edu; Tel.: +1-817-272-7467

[†] These authors contributed equally to this work.

Abstract: In this paper, we propose a novel time and control effort optimal aggressive trajectory synthesis and control design methodology. The trajectory synthesis is a modified minimum snap design but with specific position and orientation constraints on a multi-copter, such as flying through tight spaces (windows) at specific orientations. The paper also introduces a means to stitch together multiple flight segments, enforce smoothness, and minimize segment times as well as the overall time, thereby resulting in very aggressive and feasible trajectories. A novel analysis for a specific scenario when no yaw angle specifications are provided is conducted, wherein a trade-off results in additional aggressiveness. The control algorithms to follow these trajectories are based on an inverse dynamics approach. Several candidate high-fidelity simulations are performed to verify the effectiveness of the proposed approach.

Keywords: multi-copter; aggressive; optimal trajectory; constraints; nonlinear control



Citation: Liu, T.-L.; Subbarao, K. Optimal Aggressive Constrained Trajectory Synthesis and Control for Multi-Copters. *Aerospace* **2022**, *9*, 281. <https://doi.org/10.3390/aerospace9060281>

Academic Editor: Mostafa Hassanalian

Received: 11 April 2022

Accepted: 17 May 2022

Published: 24 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multi-copters, such as quadcopters, hexacopters, octocopters, and others have been very popular in recent years mainly because of their availability, mobility, agility, and flexibility. They are a great platform for control experiments and various applications. Their characteristics also make them an attractive choice for high-speed aerial navigation through complex environments. Various researches related to aggressive trajectory generation and aggressive maneuver tracking have been conducted. Some recent examples include [1], in which the authors proposed a novel control law for accurate tracking of aggressive quadcopter trajectories. In [2] the authors presented a framework to do optimal time allocation for quadcopter trajectory generation. In [3] the authors addressed the problem of performing aggressive quadcopter maneuvers that are attitude-constrained.

Our main contribution in this paper is the development of a synthesis framework that fully utilizes the available dynamic capability of the multi-copter when performing the most aggressive maneuver. We pursue aggressiveness because time is a critical issue in the given scenario. We adopt aggressive maneuvers because the environment is complicated and sometimes very specific maneuvers are needed in order to satisfy path and vehicle state constraints, given the environment model. However, the level of aggressiveness one can achieve depends on the optimality of the trajectory plan and the dynamic capability of the vehicle. It will be valuable if one can generate a trajectory that fully utilizes the dynamic capability of the vehicle while accommodating the maneuvers required at specific locations. Therefore, in this research, we assume that the dynamic capability of the multi-copter is limited by its maximum rotor thrust and define the aggressiveness accordingly.

For multi-copter trajectory generation and motion control, ideas such as multi-segment polynomial, minimum derivative optimization, and differential flatness inverse dynamics analysis are widely adopted [4–6]. To achieve an aggressive trajectory, methods such as segment time allocation [7] and spatial-temporal trajectory [2] have been proposed. However,

we do not yet have a synthetic architecture to find the most aggressive trajectory based on the waypoints, the requirement for vehicle heading and maneuvering at waypoints, and the constraints of rotor thrust for a multi-copter. We would further like to emphasize that the contribution of this research lies in the development of a complete synthesis to fulfill this gap. The constraint on vehicle maneuvering at waypoints is posed as a window passing through problem and solved through the trajectory optimization as trajectory derivative constraints. The maximum rotor thrust for the trajectory is obtained through the inverse dynamics analysis. As the optimized aggressive trajectory is obtained, a geometric controller is used to perform the polynomial trajectory tracking and verify the feasibility of the trajectory and the accuracy of the inverse dynamics analysis. Further, a novel method for yaw trajectory optimization is also proposed to further improve the aggressive performance in the case when there are no specific requirements on the vehicle heading.

The rest of the paper is organized as follows. Section 2 describes the problem and the setup. Section 3 summarizes the mathematical modeling providing the basic governing equations. Section 4 discusses the solution methodology in detail, followed by the numerical results in Section 5. Section 6 summarizes the performance of the geometric controller with Section 7 providing details on the modification to the trajectory should there be no heading constraints. Finally, Section 8 provides concluding remarks and summarizes future work.

2. Problem Description and Setup

While aggressive flight using multi-copters is widely discussed/mentioned, there is not much effort put toward defining aggressiveness and finding the most aggressive trajectory with complex maneuver requirements for a given vehicle model. Therefore, in this paper, the problem is addressed as: Given waypoint coordinates, heading angles, and some vehicle velocity and attitude constraints at these waypoints, how can we find a feasible trajectory that fully utilizes the dynamic capability of a multi-copter to achieve the optimal aggressiveness? A formal definition of aggressiveness will be provided in the latter sections.

3. Mathematical Model Description

A generic model for quadcopters is used in this paper. Note that the proposed approach can be applied to different configurations of multi-copters by modifying the vehicle force and moment model according to the desired rotor configuration and control allocation method. The coordinate systems and forces and moments generated by the rotors are shown in Figure 1.

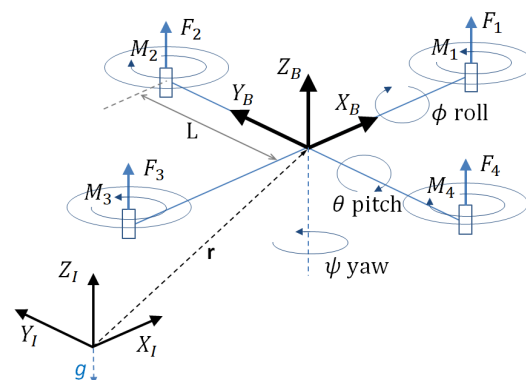


Figure 1. Reference frames and quadcopter forces/moments.

The body frame, B , is attached to the center of mass of the quadcopter and rotor 1 is on the positive X_B -axis. The gravitational acceleration g is in the $-Z_I$ direction of the inertial frame, I . Euler angles roll ϕ , pitch θ , and yaw ψ are used to define orientation from

inertial frame to body frame. Note, $Z - X - Y$ rotation order is used here, and therefore the rotation matrix for transforming coordinates from B to I is given by:

$$\mathbf{R}_{IB} = \begin{bmatrix} \cos \psi \cos \theta - \sin \phi \sin \psi \sin \theta & -\cos \phi \sin \psi & \cos \psi \sin \theta + \cos \theta \sin \phi \sin \psi \\ \cos \theta \sin \psi + \cos \psi \sin \phi \sin \theta & \cos \phi \cos \psi & \sin \psi \sin \theta - \cos \psi \cos \theta \sin \phi \\ -\cos \phi \sin \theta & \sin \phi & \cos \phi \cos \theta \end{bmatrix} \quad (1)$$

The position vector of the quadcopter in the inertial frame is denoted by \mathbf{r} . With the gravity force acting in the $-Z_I$ direction and the forces of the rotors acting in the Z_B direction, the equation governing the acceleration of the quadcopter with respect to inertial frame is given by:

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathbf{R}_{IB} \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} \quad (2)$$

With p , q , and r denoting the components of angular velocity of the quadcopter in the body frame, the rotational kinematics equation is given by:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\cos \phi \sin \theta \\ 0 & 1 & \sin \phi \\ \sin \theta & 0 & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (3)$$

Assuming rotors 1 and 3 rotate in the $-Z_B$ direction while 2 and 4 rotate in the Z_B direction, M_1 and M_3 act in the Z_B direction while M_2 and M_4 act in the $-Z_B$ direction since the moment produced by the rotor is opposite to the direction of rotation of the blade. With I denoting the moment of inertia of the quadcopter referenced to the center of mass and L denoting the distance from the axis of rotation of the rotors to the center of the quadcopter, the rotational dynamics equation is given by:

$$\mathbf{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (4)$$

Define the input $u = [u_1 \ u_2 \ u_3 \ u_4]^T$ wherein u_1 is the total force from the rotors and u_2 , u_3 and u_4 are the moments about X_B , Y_B and Z_B axes. Following [6], we assume that the force and moment produced by the i th rotor are proportional to the square of its rotational speed as:

$$F_i = k_F \omega_i^2, \quad M_i = k_M \omega_i^2 \quad (5)$$

The relationship between the input and the angular speed of the rotors can be represented as:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} F_1 + F_2 + F_3 + F_4 \\ L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (6)$$

4. Solution Methodology

We use multi-segment polynomials to generate the trajectory and assure its smoothness by having trajectory derivative continuities. To optimize the polynomial coefficients, we form the quadratic problem with the cost on the trajectory derivative which is related to the actuator input of the vehicle. We accommodate waypoint velocity and attitude requirements in the optimization problem as equality constraints. To optimize the segment time between waypoints, we introduce an augmented cost on both trajectory derivative and total time. We connect the polynomial trajectory and actual quadcopter dynamics and rotor inputs by performing inverse dynamics analysis. We finally define aggressiveness and

find the optimal trajectory with desired aggressiveness by tracking the desired maximum rotor force needed for the optimized trajectory. The whole methodology and the solution procedure are illustrated below in Figure 2.

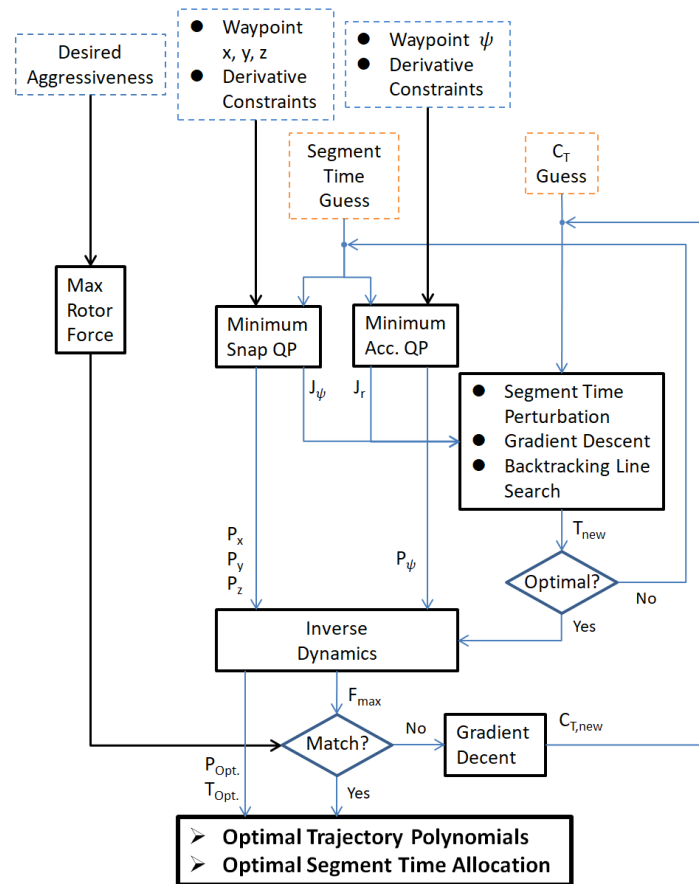


Figure 2. Methodology of the aggressive trajectory optimization.

4.1. Multi-Segment Polynomial Trajectory Optimization

Multi-segment polynomials are used to generate the trajectory. For each segment connecting one and the subsequent waypoint, independent polynomials are used to represent the quadcopter states x , y , z , and ψ (yaw angle). Each polynomial segment is represented as:

$$P(t) = p_N t^N + p_{N-1} t^{N-1} + \dots + p_0 = \sum_{i=0}^N p_i t^i \quad (7)$$

The cost function on the integral of the quadratic of the trajectory derivatives is:

$$J = \int_0^T [c_0 P(t)^2 + c_1 P'(t)^2 + c_2 P''(t)^2 + \dots + c_N P^{(N)}(t)^2] dt \quad (8)$$

where T is the flight time for the trajectory segment and c_r is user-specified penalty on the r^{th} derivative of the trajectory. This function can be written in matrix form as:

$$J = \bar{p}^T Q \bar{p} \quad (9)$$

where $\bar{p} = [p_0 \ p_1 \ \cdots \ p_N]^T$ and Q is the cost matrix. Following the formulation in [4], Q can be constructed as:

$$\text{for } r = 0, 1, \dots, N, \ i = 0, 1, \dots, N, \ l = 0, 1, \dots, N$$

$$Q_r(i+1, l+1) = \begin{cases} [\prod_{m=0}^{r-1} (i-m)(l-m)] \frac{T^{i+l-2r+1}}{i+l-2r+1} & \text{if } i \geq r \wedge l \geq r \\ 0 & \text{if } i < r \vee l < r \end{cases} \quad (10)$$

$$Q = \sum_{r=0}^N c_r Q_r \quad (11)$$

In this paper, to minimize the input needed to achieve optimal aggressiveness, we adopted the choice in [5] which is to minimize the snap (4th derivative) in trajectory x , y , and z while minimizing the acceleration (2nd derivative) in trajectory ψ . Therefore, we use $c_4 = 1$ while all other coefficients are set to zero for x , y , and z polynomial, and only $c_2 = 1$ for yaw polynomial. The constraints on the derivatives on the endpoints of a polynomial segment can be imposed as a linear function of the coefficients:

$$A\bar{p} = b$$

$$A = \begin{bmatrix} A_0 \\ A_T \end{bmatrix}, \quad b = \begin{bmatrix} b_0 \\ b_T \end{bmatrix} \quad (12)$$

where A is constructed by evaluating the components in the derivative formulations of the polynomial at $t = 0$ and $t = T$ corresponding to the appropriate coefficients as:

$$A_{0rn} = \begin{cases} \prod_{m=0}^{r-1} (n-m) & \text{if } r = n \\ 0 & \text{if } r \neq n \end{cases} \quad (13)$$

$$A_{Trn} = \begin{cases} [\prod_{m=0}^{r-1} (n-m)] T^{n-r} & \text{if } r \leq n \\ 0 & \text{if } r > n \end{cases} \quad (14)$$

We use the 0th order derivative constraint to specify the waypoint position. Higher order derivatives can be used to specify desired waypoint velocity, acceleration, etc., e.g., to enforce that the quadcopter starts from rest at the beginning of a trajectory. If not specified, these derivatives are subject to minimization of the cost function. Having assembled Q , A , and b , the quadratic problem can be written below. There are methods to solve such a standard equality constrained QP [8]. In this research, the Matlab solver “quadprog” is used.

$$\begin{aligned} \min_{\bar{p}} \quad & \bar{p}^T Q \bar{p} \\ \text{s.t.} \quad & A\bar{p} - b = 0 \end{aligned} \quad (15)$$

For M polynomial segments, the joint optimization can be composed by concatenating their cost matrices in a block-diagonal fashion as:

$$J_{\text{joint}} = \begin{bmatrix} \bar{p}_1 \\ \vdots \\ \bar{p}_M \end{bmatrix}^T \begin{bmatrix} Q_1(T_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & Q_M(T_M) \end{bmatrix} \begin{bmatrix} \bar{p}_1 \\ \vdots \\ \bar{p}_M \end{bmatrix} \quad (16)$$

The derivative constraints can also be concatenated in a block-diagonal fashion as:

$$\begin{bmatrix} A_1(T_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & A_M(T_M) \end{bmatrix} \begin{bmatrix} \bar{p}_1 \\ \vdots \\ \bar{p}_M \end{bmatrix} = A_{\text{der}} \begin{bmatrix} \bar{p}_1 \\ \vdots \\ \bar{p}_M \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix} \quad (17)$$

To have a feasible smooth trajectory, the derivatives should be continuous between segments. The continuity constraints must be imposed to ensure that the derivatives at the end of the i th segment match the derivatives at the beginning of the $(i + 1)$ th segment:

$$A_{T,i}\bar{p}_i = A_{0,i+1}\bar{p}_{i+1} \quad (18)$$

$$\begin{bmatrix} A_{T,1} & -A_{0,2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & A_{T,2} & -A_{0,3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & A_{T,M-1} & -A_{0,M} \end{bmatrix} \begin{bmatrix} \bar{p}_1 \\ \bar{p}_2 \\ \bar{p}_3 \\ \vdots \\ \bar{p}_{M-1} \\ \bar{p}_M \end{bmatrix} = A_{\text{con}} \begin{bmatrix} \bar{p}_1 \\ \bar{p}_2 \\ \bar{p}_3 \\ \vdots \\ \bar{p}_{M-1} \\ \bar{p}_M \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad (19)$$

These constraints can be compiled into a single set of linear equality constraints for the joint optimization problem:

$$\begin{bmatrix} A_{\text{der}} \\ A_{\text{con}} \end{bmatrix} \begin{bmatrix} \bar{p}_1 \\ \vdots \\ \bar{p}_M \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_M \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (20)$$

4.2. Constraints on Vehicle Velocity and Attitude at Waypoints

In scenarios such that the quadcopter needs to pass through some narrow gap (e.g., a window) as shown in Figure 3, we can utilize the derivative constraints to achieve the objective.

Assume a waypoint at the center of the window and the window orientation is defined with $Z - X - Y$ Euler angles roll ϕ , pitch θ , and yaw ψ . The window forward vector in inertial frame can be obtained as:

$$w_F = R_{IB} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \psi \cos \theta - \sin \phi \sin \psi \sin \theta \\ \cos \theta \sin \psi + \cos \psi \sin \phi \sin \theta \\ -\cos \phi \sin \theta \end{bmatrix} = \begin{bmatrix} W_{F_x} \\ W_{F_y} \\ W_{F_z} \end{bmatrix} \quad (21)$$

The window upward vector in inertial frame can be obtained as:

$$w_U = R_{IB} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \psi \sin \theta + \cos \theta \sin \phi \sin \psi \\ \sin \psi \sin \theta - \cos \psi \cos \theta \sin \phi \\ \cos \phi \cos \theta \end{bmatrix} = \begin{bmatrix} W_{U_x} \\ W_{U_y} \\ W_{U_z} \end{bmatrix} \quad (22)$$

To fly through the window safely, the velocity vector at the waypoint should be aligned with the window forward vector. Therefore, the cross product of the vectors should be zero, and the constraints for the waypoint 1st order derivatives can be composed as:

$$v \times w_F = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \times \begin{bmatrix} W_{F_x} \\ W_{F_y} \\ W_{F_z} \end{bmatrix} = \begin{bmatrix} \dot{y}W_{F_z} - \dot{z}W_{F_y} \\ \dot{z}W_{F_x} - \dot{x}W_{F_z} \\ \dot{x}W_{F_y} - \dot{y}W_{F_x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (23)$$

From the equation of acceleration, we can deduce that z_B in inertial frame is in the direction of the vector $[\ddot{x} \quad \ddot{y} \quad \ddot{z} + g]^T$:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R_{IB} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \sum F \Rightarrow m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} + g \end{bmatrix} = z_B \cdot \sum F \quad (24)$$

For safely flying through the window, the z_B vector of the quadcopter should be aligned with the window upward vector. Therefore, the constraints for the waypoint 2nd order derivatives can be composed as:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} + g \end{bmatrix} \times \begin{bmatrix} W_{U_x} \\ W_{U_y} \\ W_{U_z} \end{bmatrix} = \begin{bmatrix} \ddot{y}W_{U_z} - (\ddot{z} + g)W_{U_y} \\ (\ddot{z} + g)W_{U_x} - \ddot{x}W_{U_z} \\ \ddot{x}W_{U_y} - \ddot{y}W_{U_x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (25)$$

$$\Rightarrow \begin{bmatrix} \ddot{y}W_{U_z} - \ddot{z}W_{U_y} \\ \ddot{z}W_{U_x} - \ddot{x}W_{U_z} \\ \ddot{x}W_{U_y} - \ddot{y}W_{U_x} \end{bmatrix} = \begin{bmatrix} gW_{U_y} \\ -gW_{U_x} \\ 0 \end{bmatrix} \quad (26)$$

For these constraints, the relationship between x , y , and z derivatives needs to be specified. Therefore, the joint optimization of x , y , and z should be composed by concatenating their cost and constraint matrices in a block-diagonal fashion as:

$$J_{xyz} = \begin{bmatrix} \bar{p}_x \\ \bar{p}_y \\ \bar{p}_z \end{bmatrix}^T \begin{bmatrix} Q_x & 0 & 0 \\ 0 & Q_y & 0 \\ 0 & 0 & Q_z \end{bmatrix} \begin{bmatrix} \bar{p}_x \\ \bar{p}_y \\ \bar{p}_z \end{bmatrix} \quad (27)$$

$$\begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} \bar{p}_x \\ \bar{p}_y \\ \bar{p}_z \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (28)$$

where $\bar{p}_x = [\bar{p}_{x_1} \cdots \bar{p}_{x_M}]^T$, $\bar{p}_y = [\bar{p}_{y_1} \cdots \bar{p}_{y_M}]^T$ and $\bar{p}_z = [\bar{p}_{z_1} \cdots \bar{p}_{z_M}]^T$. With this joint optimization setup, the constraints for a window passage on waypoint s in an M -segment trajectory can be implemented as:

$$\begin{bmatrix} 0 & W_{F_z}v_{0,s} & -W_{F_y}v_{0,s} \\ -W_{F_z}v_{0,s} & 0 & W_{F_x}v_{0,s} \\ W_{F_y}v_{0,s} & -W_{F_x}v_{0,s} & 0 \\ 0 & W_{U_z}a_{0,s} & -W_{U_y}a_{0,s} \\ -W_{U_z}a_{0,s} & 0 & W_{U_x}a_{0,s} \\ W_{U_y}a_{0,s} & -W_{U_x}a_{0,s} & 0 \end{bmatrix} \begin{bmatrix} \bar{p}_x \\ \bar{p}_y \\ \bar{p}_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ gW_{U_y} \\ -gW_{U_x} \\ 0 \end{bmatrix} \quad (29)$$

$$\begin{cases} v_{0,s} = \begin{bmatrix} [0 \times (n+1)] \times (s-1) & v_0 & [0 \times (n+1)] \times (M-s) \end{bmatrix} \\ a_{0,s} = \begin{bmatrix} [0 \times (n+1)] \times (s-1) & a_0 & [0 \times (n+1)] \times (M-s) \end{bmatrix} \end{cases} \quad (30)$$

For a segment polynomial $P(t) = p_0 + p_1t^1 + p_2t^2 + p_3t^3 + \cdots + p_Nt^N$, the 1st order derivative (velocity) at $t = 0$ is $v_0\bar{p}$ and $v_0 = [0 \ 1 \ 0 \ 0 \ \cdots \ 0]$. The 2nd order derivative (acceleration) at $t = 0$ is $a_0\bar{p}$ and $a_0 = [0 \ 0 \ 2 \ 0 \ \cdots \ 0]$.

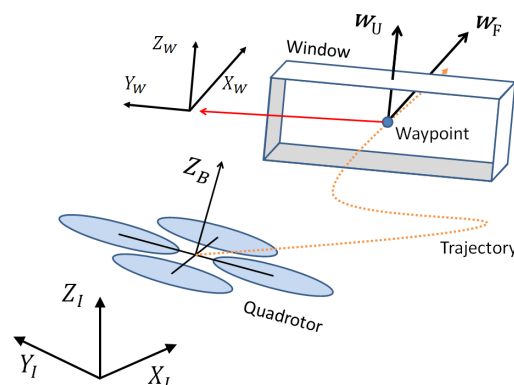


Figure 3. Reference frames and quadcopter forces/moments.

4.3. Segment Time Optimization

The above optimization finds the optimal trajectory polynomials for the given segment times. However, in general cases, we do not specify segment times, and instead, we would like to also optimize them to achieve better aggressiveness. To optimize segment times, we modify the cost function in the form:

$$J_T = J_{org} + c_T \sum_{i=1}^M T_i \quad (31)$$

where J_{org} combines the original cost on x , y , z , and ψ while c_T is a penalty on total time. The cost on x , y , z is the integral of square of derivatives on distance while the cost on ψ is on angle. To combine the costs, two coefficients μ_r and μ_ψ are introduced to non-dimensionalize the costs.

$$J_{org} = \mu_r J_{xyz} + \mu_\psi J_\psi \quad (32)$$

In this research, we minimize the 4th derivative in trajectory x , y , and z while minimizing the 2nd derivative in trajectory yaw angle. Therefore, the above equation can be represented as:

$$J_{org} = \mu_r \int_0^{T_f} \left[\left(\frac{d^4 x}{dt^4} \right)^2 + \left(\frac{d^4 y}{dt^4} \right)^2 + \left(\frac{d^4 z}{dt^4} \right)^2 \right] dt + \mu_\psi \int_0^{T_f} \left(\frac{d^2 \psi}{dt^2} \right)^2 dt \quad (33)$$

where μ_r and μ_ψ are defined as:

$$\mu_r = \left[\frac{1}{\max \left(\left| \frac{d^4 x}{dt^4} \right|, \left| \frac{d^4 y}{dt^4} \right|, \left| \frac{d^4 z}{dt^4} \right| \right)} \right]^2 \quad (34)$$

$$\mu_\psi = \left[\frac{1}{\max \left(\left| \frac{d^2 \psi}{dt^2} \right| \right)} \right]^2 \quad (35)$$

We find μ_r and μ_ψ in the first iteration with initial segment times and then use them as constants throughout the optimization process. With the cost function defined, we perturb each segment time by some δt to obtain the gradient of the cost function with respect to each segment time. This is then used in a gradient descent method to find the time allocation for the minimum cost iteratively.

$$\bar{T} = [T_1 \quad \cdots \quad T_M]^T \quad (36)$$

$$\nabla_i J_T = \frac{J_T [T_1 \quad \cdots \quad T_i + \delta t \quad \cdots \quad T_M]^T - J_T(\bar{T})}{\delta t} \quad (37)$$

$$T_i^{new} = T_i + \alpha \Delta T_i, \quad \Delta T_i = - \frac{\nabla_i J_T}{\|\nabla_i J_T\|} \quad (38)$$

Since this is a high-dimensional problem with a complex cost function, the step size α can easily become too small or too large during the iterations and lead to slow convergence or even divergence. For numerical efficiency, stability, and convergence, we use the backtracking line search method [9] to find a suitable step size α in every iteration.

$$\text{while } J(\bar{T} + \alpha \Delta \bar{T}) > J(\bar{T}) + \epsilon \alpha \nabla_i J_T^T \Delta \bar{T}, \quad \alpha := \beta \alpha \quad (39)$$

With $\beta = 0.5$ and $\epsilon = 0.0001$, we have fast and stable convergence in our test cases. Figure 4 shows the result of segment time optimization for a simple 2D 4-waypoint scenario.

The initial segment time is $\bar{T} = [1 \ 3 \ 1]^T$ s, and we optimize it with two different c_T values, 50 and 5000. The result trajectories look identical because they have similar segment time distribution while the total time turned out to be 6.28 and 3.53 seconds, respectively. It is observed that, with this segment time optimization process, the optimal segment time distribution will be found to minimize the cost on integral of the square of trajectory derivatives, while the optimal total time is found based on the value of c_T . The larger c_T used, the smaller the total time will be, which makes the trajectory more aggressive.

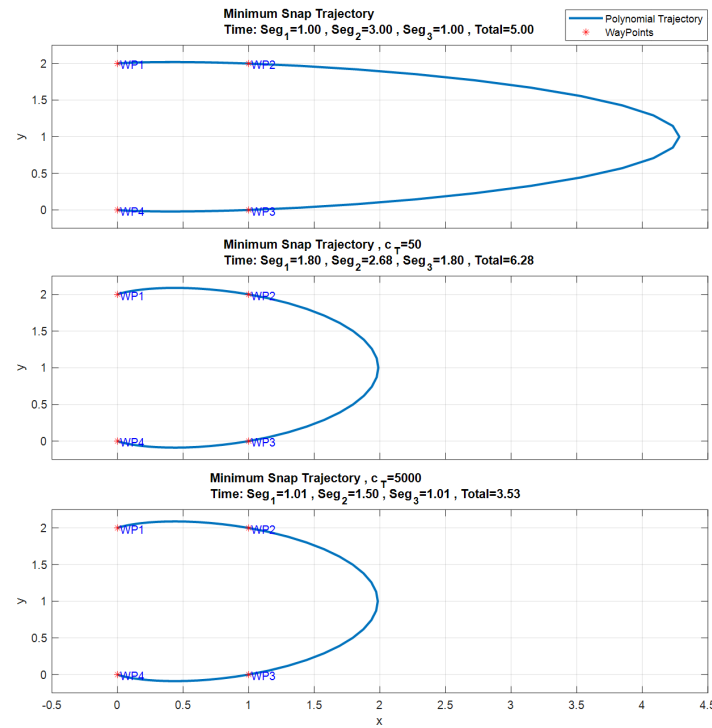


Figure 4. Comparison of trajectories optimized with different c_T .

4.4. Inverse Dynamics

To connect the polynomial trajectory optimization framework and actual quadcopter dynamics, we perform the inverse dynamics analysis. The differential flatness method is widely adopted in inverse dynamics analysis for multi-copters [6]. In this process, we will find all the states and inputs of the quadcopter according to the trajectory x, y, z, ψ , and their derivatives. For orientation ϕ and θ , first from the equation of acceleration we have:

$$\mathbf{z}_B = \frac{[\ddot{x} \ \ddot{y} \ \ddot{z} + g]^T}{\|[\ddot{x} \ \ddot{y} \ \ddot{z} + g]\|}, \quad u_1 = m \|[\ddot{x} \ \ddot{y} \ \ddot{z} + g]\| \quad (40)$$

Assume \mathbf{x}_C is the vector obtained by rotating \mathbf{x}_I around \mathbf{z}_I by yaw angle ψ :

$$\mathbf{x}_C = [\cos \psi \ \sin \psi \ 0]^T \quad (41)$$

We can determine \mathbf{x}_B and \mathbf{y}_B by:

$$\mathbf{y}_B = \frac{\mathbf{z}_B \times \mathbf{x}_C}{\|\mathbf{z}_B \times \mathbf{x}_C\|}, \quad \mathbf{x}_B = \mathbf{y}_B \times \mathbf{z}_B \quad (42)$$

With vehicle body frame defined, we can determine the rotation matrix and roll and pitch angles by:

$$\mathbf{R}_{IB} = [\mathbf{x}_B \ \mathbf{y}_B \ \mathbf{z}_B] \quad (43)$$

$$\phi = \arcsin [R_{IB}(3,2)] , \quad \theta = \arctan \left[\frac{-R_{IB}(3,1)}{R_{IB}(3,3)} \right] \quad (44)$$

For angular velocity p , q , and r , first we take the 1st derivative of the equation of acceleration:

$$m\dot{\mathbf{a}} = \dot{u}_1 \mathbf{z}_B + \boldsymbol{\omega}_B \times u_1 \mathbf{z}_B \quad (45)$$

Substituting $\dot{u}_1 = \mathbf{z}_B \cdot m\dot{\mathbf{a}}$ we have:

$$\boldsymbol{\omega}_B \times \mathbf{z}_B = \frac{m}{u_1} [\dot{\mathbf{a}} - (\mathbf{z}_B \cdot \dot{\mathbf{a}}) \mathbf{z}_B] \quad (46)$$

With $\dot{\mathbf{a}} = [\ddot{x} \quad \ddot{y} \quad \ddot{z}]$, the RHS is known. With \mathbf{x}_B , \mathbf{y}_B , and \mathbf{z}_B being unit vectors, $\boldsymbol{\omega}_B \times \mathbf{z}_B$ can be considered as the projection of $\boldsymbol{\omega}_B$ onto the $\mathbf{x}_B - \mathbf{y}_B$ plane with 90° shift. Therefore, the body angular velocities p and q can be determined as:

$$p = -(\boldsymbol{\omega}_B \times \mathbf{z}_B) \cdot \mathbf{y}_B , \quad q = (\boldsymbol{\omega}_B \times \mathbf{z}_B) \cdot \mathbf{x}_B \quad (47)$$

From the rotational kinematics equation, we have:

$$\begin{aligned} \dot{\psi} &= -\frac{\sin \theta}{\cos \phi} p + \frac{\cos \theta}{\cos \phi} r \\ \Rightarrow r &= \left(\dot{\psi} + \frac{\sin \theta}{\cos \phi} p \right) \frac{\cos \phi}{\cos \theta} \end{aligned} \quad (48)$$

With p , q , and r solved, we have $\boldsymbol{\omega}_B = \mathbf{R}_{IB}[p \ q \ r]^T$, and $\dot{\theta}$ and $\dot{\phi}$ can also be obtained from the inversion of the rotational kinematics equation. For angular acceleration \dot{p} , \dot{q} and \dot{r} , first we take the 2nd derivative of the equation of acceleration:

$$\begin{aligned} m\ddot{\mathbf{a}} &= \ddot{u}_1 \mathbf{z}_B + 2\boldsymbol{\omega}_B \times \dot{u}_1 \mathbf{z}_B + \boldsymbol{\omega}_B \times \boldsymbol{\omega}_B \times u_1 \mathbf{z}_B + \dot{\boldsymbol{\omega}}_B \times u_1 \mathbf{z}_B \\ \Rightarrow \dot{\boldsymbol{\omega}}_B \times \mathbf{z}_B &= (m\ddot{\mathbf{a}} - \ddot{u}_1 \mathbf{z}_B - 2\boldsymbol{\omega}_B \times \dot{u}_1 \mathbf{z}_B - \boldsymbol{\omega}_B \times \boldsymbol{\omega}_B \times u_1 \mathbf{z}_B) / u_1 \end{aligned} \quad (49)$$

With $\ddot{\mathbf{a}} = [\ddot{x} \quad \ddot{y} \quad \ddot{z}]$, $\ddot{u}_1 = \mathbf{z}_B \cdot (m\ddot{\mathbf{a}} - \boldsymbol{\omega}_B \times \boldsymbol{\omega}_B \times u_1 \mathbf{z}_B)$ and $\dot{u}_1 = \mathbf{z}_B \cdot m\dot{\mathbf{a}}$, the RHS is known and the body angular accelerations \dot{p} and \dot{q} can be determined as:

$$\dot{p} = -(\dot{\boldsymbol{\omega}}_B \times \mathbf{z}_B) \cdot \mathbf{y}_B , \quad \dot{q} = (\dot{\boldsymbol{\omega}}_B \times \mathbf{z}_B) \cdot \mathbf{x}_B \quad (50)$$

Taking derivative of previous $\dot{\psi}$ equation, we have:

$$\begin{aligned} \ddot{\psi} &= -\frac{\sin \theta}{\cos \phi} \dot{p} + \frac{\cos \theta}{\cos \phi} \dot{r} - p \frac{d}{dt} \left(\frac{\sin \theta}{\cos \phi} \right) + r \frac{d}{dt} \left(\frac{\cos \theta}{\cos \phi} \right) \\ \Rightarrow \dot{r} &= \frac{\cos \phi}{\cos \theta} \left[\ddot{\psi} + \frac{\sin \theta}{\cos \phi} \dot{p} + p \left(\frac{\cos \theta \cos \phi \dot{\theta} + \sin \theta \sin \phi \dot{\phi}}{\cos^2 \phi} \right) \right. \\ &\quad \left. - r \left(\frac{\cos \theta \sin \phi \dot{\phi} - \sin \theta \cos \phi \dot{\theta}}{\cos^2 \phi} \right) \right] \end{aligned} \quad (51)$$

Next we determine the moment inputs from obtained angular velocity and acceleration by the rotational dynamics equation:

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \mathbf{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (52)$$

Now we have found all the states and inputs of the quadcopter derived from the trajectory x, y, z, ψ , and their derivatives. We can further determine the angular speed and force produced of each rotor by:

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}, \quad F_i = k_F \omega_i^2 \quad (53)$$

4.5. Max Force Tracking and Aggressiveness Defined

We find the maximum rotor force needed for the trajectory through the inverse dynamics analysis and observe how the maximum force and the trajectory total time vary with different c_T values. Figure 5 shows the analysis result for a 3D 4-waypoint scenario. It is observed that, with the increment in c_T , the total time decreases smoothly while the maximum force rises smoothly with increasing slope. With this relationship, we can track a particular total time or maximum force for a given scenario by the gradient descent method through c_T .

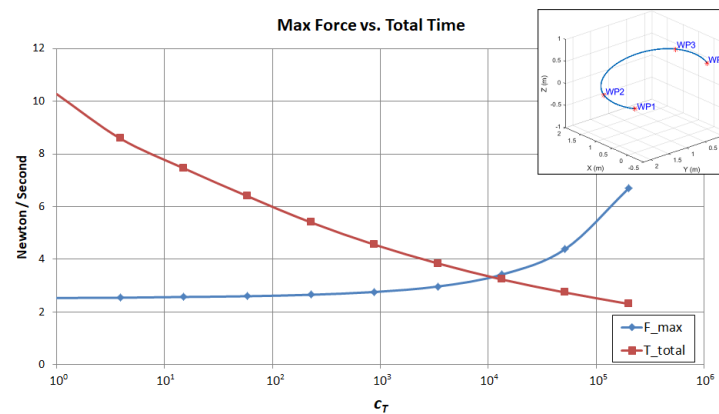


Figure 5. Comparison of maximum rotor force and total time for trajectories optimized with different c_T .

With optimal aggressiveness, the quadcopter should finish the tasks in the shortest time possible by means of its dynamic capability. Generally speaking, the dynamic capability of a quadcopter is restricted by the maximum thrust of its rotors. Therefore, we define aggressiveness as the percentage of excess thrust required for the optimized trajectory.

$$\text{Aggressiveness} = \frac{F_{\max_req} - F_{\text{hover}}}{F_{\max} - F_{\text{hover}}} \cdot 100\% \quad (54)$$

where F_{\max_req} is the maximum rotor force required for the trajectory, F_{hover} is the rotor force for steady hover, and F_{\max} is the maximum thrust of the rotor. One might conserve the aggressiveness for safety reasons. Given waypoints x, y, z, ψ , and the desired aggressiveness, the whole methodology to find the optimal trajectory and segment time allocation for a given quadcopter model is summarized in Figure 2.

5. Numerical Results

The quadcopter parameters we use in this research are from [10] and tabulated below in Table 1.

Table 1. Quadcopter parameters.

m	1.023 kg	g	9.81 m/s ²
L	0.2223 m	I_{xx}	0.0095 kg · m ²
k_F	$1.4865 \cdot 10^{-7}$ N/RPM ²	I_{yy}	0.0095 kg · m ²
k_M	$2.9250 \cdot 10^{-9}$ N · m/RPM ²	I_{zz}	0.0186 kg · m ²

We test the optimization framework with a four-waypoint scenario. The waypoint settings are tabulated below (yaw angles at waypoints 2 and 3 are not specified) in Table 2.

Table 2. Waypoint settings in the scenario.

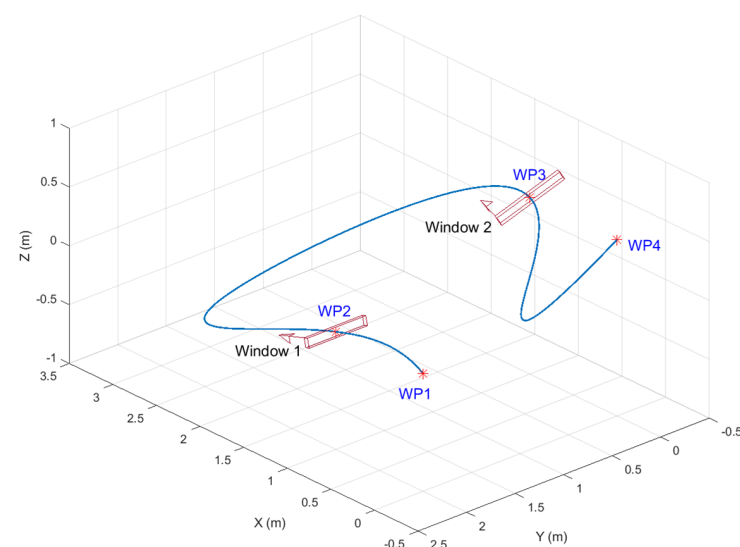
WPT Setting	x (m)	y (m)	z (m)	ψ
Waypoint 1	0	2	0	0
Waypoint 2	1	2	0	-
Waypoint 3	1	0	0.5	-
Waypoint 4	0	0	0.5	−180°

We specify the mission to be from rest to rest, therefore the velocity and acceleration at the first and the last waypoint are constrained to be zero. There are two narrow windows to pass through in this scenario. The window settings are tabulated below in Table 3.

Table 3. Window (WDW) settings in the scenario.

WDW Setting	ϕ	θ	ψ	Location
Window 1	0°	15°	0°	Waypoint 2
Window 2	−30°	0°	−20°	Waypoint 3

We specify the aggressiveness to be 80%. For this quadcopter model the rotor force for steady hover is $F_{\text{hover}} = 2.5$ N. Assuming $F_{\text{max}} = 3.75$ N, we have the maximum rotor force we can use for the trajectory as $F_{\text{max_req}} = 3.5$ N. With initial guess of $\bar{T} = [5 \ 5 \ 5]^T$ s and $c_T = 100$, we use 10th order segment polynomials and have derivative continuity constraints on up to 6th order derivative (Pop). Figure 6 shows the scenario and the optimal trajectory.

**Figure 6.** Plot of the test scenario and the optimal trajectory.

Figures 7–9 show the result of the inverse dynamics analysis from the trajectory with respect to the particular quadcopter model. The magenta dashed lines mark the times of

waypoint passages. It can be noticed that the quadcopter had a spike in angular acceleration and angular rate around waypoint 3 because it was making the maneuver to pass through the highly tilted window 2.

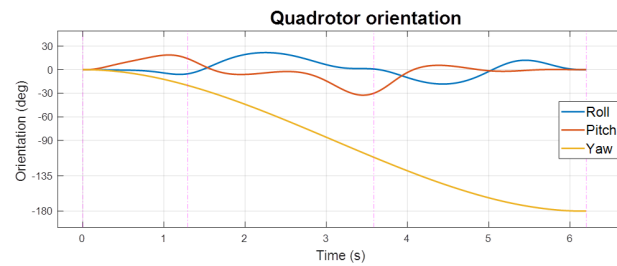


Figure 7. Vehicle Euler angles along the trajectory flight.

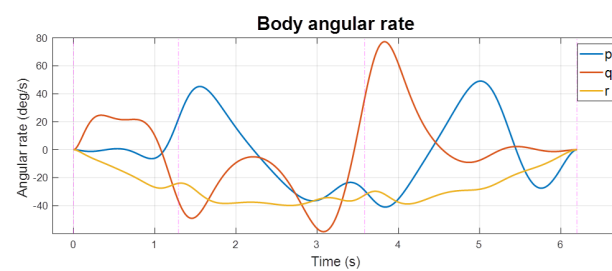


Figure 8. Vehicle body angular rate along the trajectory flight.

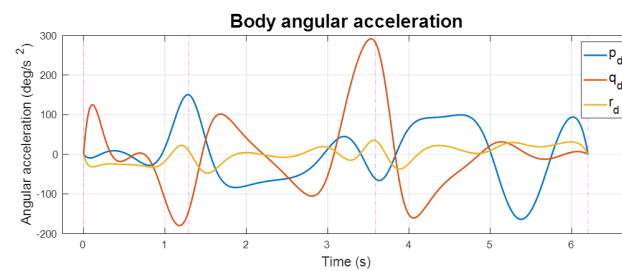


Figure 9. Vehicle body angular acceleration along the trajectory flight.

Figures 10 and 11 show the control input and the rotor thrust along the trajectory. From the plot, we confirmed that the maximum rotor thrust used is 3.5 N as expected.

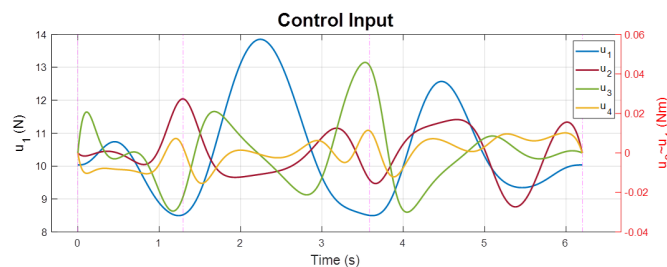


Figure 10. Control input along the trajectory flight.

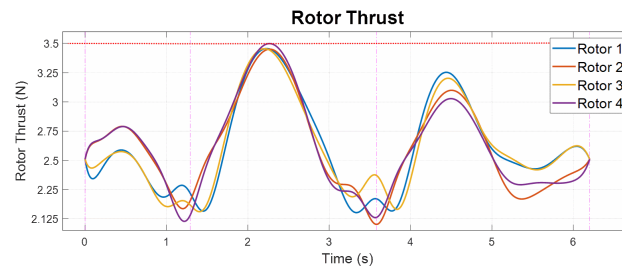


Figure 11. Rotor thrust along the trajectory flight.

Figure 12 shows the quadcopter on the trajectory with a fixed time interval of 0.33 s. It can be observed that the quadcopter had the highest speed in segment 2 so it can maneuver through the highly tilted window 2. The successful passage through the narrow windows can be visually confirmed. If we further check the data at waypoint 3, we have the window forward and upward vectors as $w_F = [0.939 \quad -0.342 \quad 0]^T$ and $w_U = [0.171 \quad 0.469 \quad 0.866]^T$. Additionally, the quadcopter velocity and thrust vectors (aligned with z_B) are $v = [-1.777 \quad 0.6470 \quad 0]^T$ and $f = [1.454 \quad 3.995 \quad 7.364]^T$. The velocity and thrust vectors are confirmed to be aligned with the window forward and upward vectors. Finally, the optimal time allocation was found as $\bar{T}_{opt} = [1.29 \quad 2.29 \quad 2.62]^T$ s.

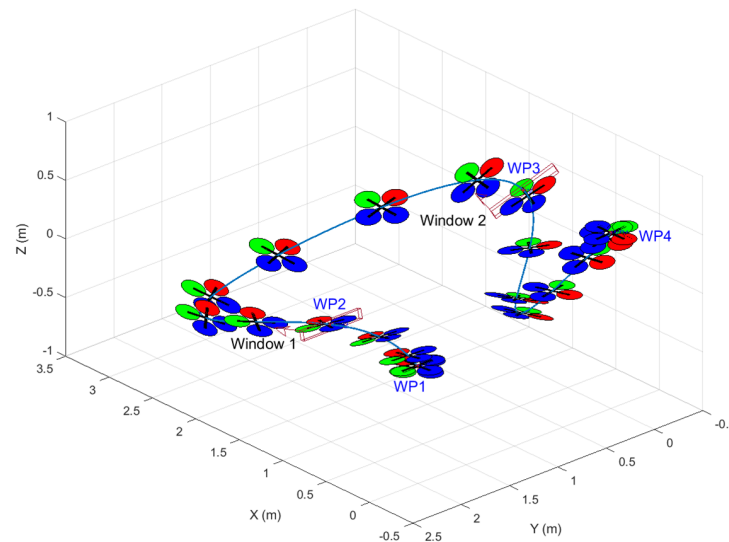


Figure 12. Plot of the quadcopter along the trajectory flight.

6. Geometric Control and Simulation Result

To perform the polynomial trajectory tracking and verify the result of the aggressive trajectory optimization, we adopted the geometric controller proposed by Lee et al. in [11] and implemented it in MATLAB/Simulink. The controller is constructed in two parts, trajectory tracking and attitude tracking. The control input of total force f is obtained by the trajectory tracking part with:

$$f = -(-k_x e_x - k_v e_v - m g e_3 + m \ddot{x}_d) \cdot R e_3 \quad (55)$$

The control input of the moments is obtained by the attitude tracking part with:

$$M = -k_R e_R - k_\omega e_\omega + \omega \times I \omega - I(\dot{\omega} R^T R_d \omega_d - R^T R_d \dot{\omega}_d) \quad (56)$$

where vector e_3 defines the direction of gravity and R is the rotation matrix from body to inertial frame. The tracking errors for position e_x , velocity e_v , attitude e_R , and angular velocity e_ω are defined as:

$$\begin{aligned} e_x &= x - x_d \\ e_v &= v - v_d \end{aligned} \quad (57)$$

$$\begin{aligned} e_R &= \frac{1}{2}(R_d^T R - R^T R_d)^\vee \\ e_\omega &= \omega - R^T R_d \omega_d \end{aligned} \quad (58)$$

where the *hat map* $\hat{\cdot} : \mathbb{R}^3 \rightarrow \text{SO}(3)$ is defined by the condition that $\hat{x}y = x \times y$ for all $x, y \in \mathbb{R}^3$. Additionally, the *vee map* $^\vee : \text{SO}(3) \rightarrow \mathbb{R}^3$ is the inverse of the hat map. At a given moment, x , v , ω , and R represent the position, linear velocity, body angular velocity, and rotation matrix of the vehicle. The corresponding desired position x_d and velocity v_d are captured from the polynomial trajectory. The desired body z axis can be obtained as:

$$z_d = -\frac{-k_x e_x - k_v e_v - m g e_3 + m \ddot{x}_d}{\| -k_x e_x - k_v e_v - m g e_3 + m \ddot{x}_d \|} \quad (59)$$

With the desired body z axis, yaw angle, and derivatives of the polynomial trajectory available, we can find the desired rotation matrix R_d , angular velocity ω_d , and angular acceleration $\dot{\omega}_d$ by following the process discussed previously in inverse dynamics. The simulation is implemented with the following equations of motion:

$$\begin{aligned} \dot{x} &= v \\ m\dot{v} &= m g e_3 - f R e_3 \end{aligned} \quad (60)$$

$$\begin{aligned} \dot{R} &= R \hat{\omega} \\ I\dot{\omega} + \omega \times I\omega &= M \end{aligned} \quad (61)$$

With 10% error in estimated inertia, ± 1 N noise added to f and ± 0.003 Nm noise added to M (around 10% of the maximum input used) as control input disturbance, we have the simulation result shown in figures below.

From the simulation result we can see that, despite the existence of inertia estimation error and control input disturbance, the geometric control had a good performance in tracking the polynomial trajectory. Position and yaw angle are well tracked and roll and pitch angles turned out to be very close to the inverse dynamics estimation as shown in Figure 13 and 14. Therefore, the successful passage through the narrow windows is confirmed. Figure 15 and 16 show that the linear acceleration is well tracked and the angular acceleration is very close to the estimation. From Figure 17, the control input used to track this polynomial trajectory is verified to be very close to the inverse dynamics estimation. The maximum rotor thrust used in the simulation is also confirmed to be around the maximum rotor force (3.5 N) requirement we specified in the trajectory generation as shown in Figure 18.

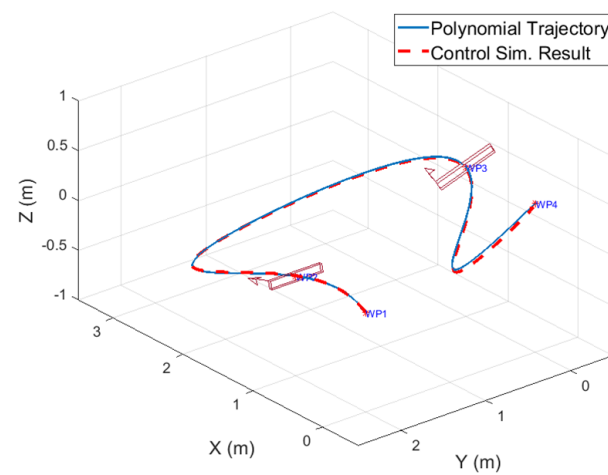


Figure 13. Trajectory tracking result.

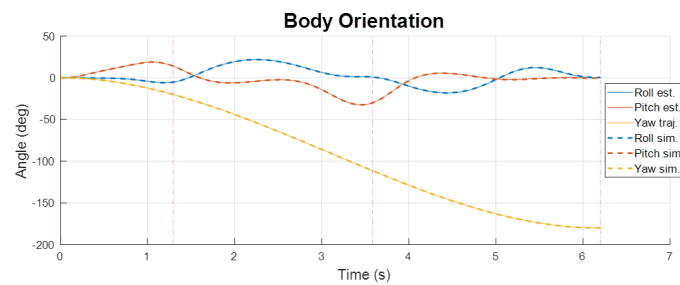


Figure 14. Vehicle Euler angle comparison.

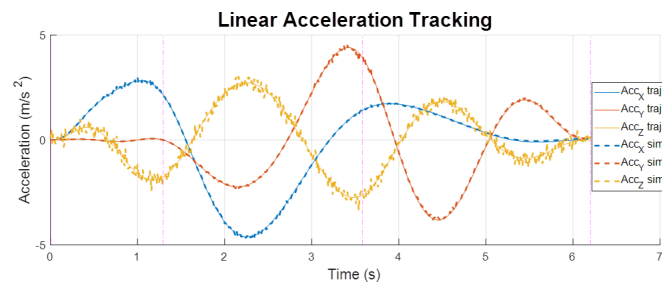


Figure 15. Vehicle linear acceleration tracking.

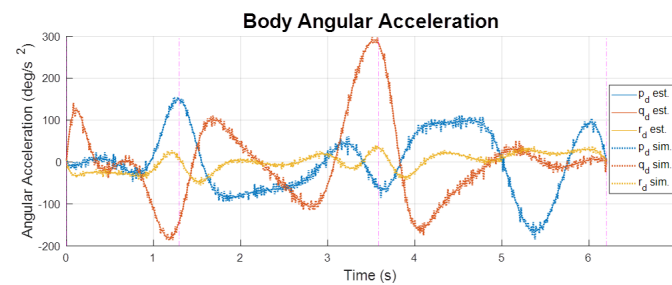


Figure 16. Vehicle angular acceleration comparison.

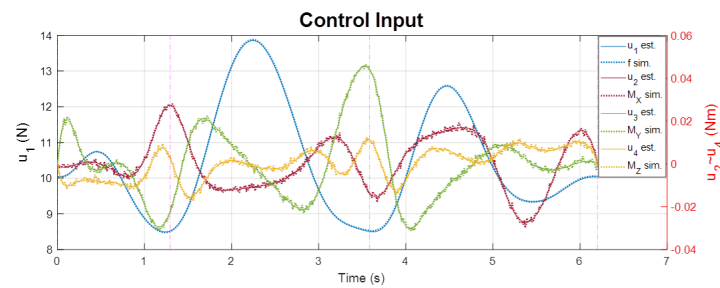


Figure 17. Control input comparison.

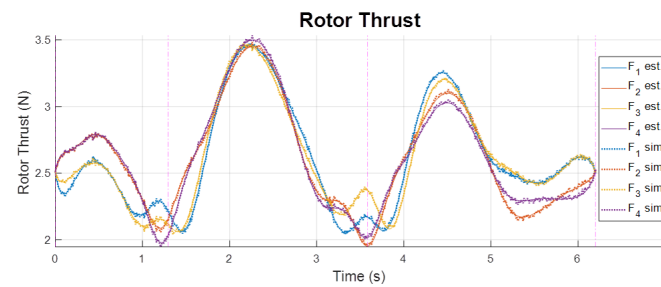


Figure 18. Rotor force comparison.

7. Yaw Trajectory Optimization

If we do not have a specific requirement on yaw angle, can we optimize the yaw trajectory to achieve better aggressive performance? In other words, by finding the yaw trajectory such that no M_z control moment input is required to track the aggressive polynomial trajectory, we can further reduce the maximum rotor force needed and thereby track the aggressive trajectory with less motor force or achieve a faster trajectory with the same aggressiveness specified. From the equations of motion:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \mathbf{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (62)$$

Because of the symmetry, the moment of inertia of the quadcopter is assumed as:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \text{ and } I_{xx} = I_{yy} \quad (63)$$

$$M_z = I_{zz}\dot{r} + pq(I_{yy} - I_{xx}) = I_{zz}\dot{r} \quad (64)$$

Therefore, to make $M_z = 0$, we need $\dot{r} = 0$. Assuming the initial state $r_0 = 0$, this means $\dot{r} = r = 0$ for the whole trajectory. Based on this, we can modify the inverse dynamics analysis to solve for $\dot{\psi}$ and $\ddot{\psi}$ as:

$$\begin{aligned} r &= \left(\dot{\psi} + \frac{\sin \theta}{\cos \phi} p \right) \frac{\cos \phi}{\cos \theta} = 0 \\ \Rightarrow \dot{\psi} &= -\frac{\sin \theta}{\cos \phi} p \end{aligned} \quad (65)$$

$$\begin{aligned} \dot{r} &= \frac{\cos \phi}{\cos \theta} \left[\ddot{\psi} + \frac{\sin \theta}{\cos \phi} \dot{p} + p \left(\frac{\cos \theta \cos \phi \dot{\theta} + \sin \theta \sin \phi \dot{\phi}}{\cos^2 \phi} \right) \right. \\ &\quad \left. - r \left(\frac{\cos \theta \sin \phi \dot{\phi} - \sin \theta \cos \phi \dot{\theta}}{\cos^2 \phi} \right) \right] = 0 \\ \Rightarrow \ddot{\psi} &= -\frac{\sin \theta}{\cos \phi} \dot{p} - p \left(\frac{\cos \theta \cos \phi \dot{\theta} + \sin \theta \sin \phi \dot{\phi}}{\cos^2 \phi} \right) \end{aligned} \quad (66)$$

Having $\dot{\psi}$ and $\ddot{\psi}$ at each moment in the trajectory, we can integrate ψ iteratively throughout the inverse dynamics analysis by $\psi_{n+1} = \psi_n + \dot{\psi}\Delta t + \frac{1}{2}\ddot{\psi}\Delta t^2$. Figure 19 and 20 show a simple four-waypoint aggressive trajectory and the corresponding optimal yaw trajectory obtained by this method.

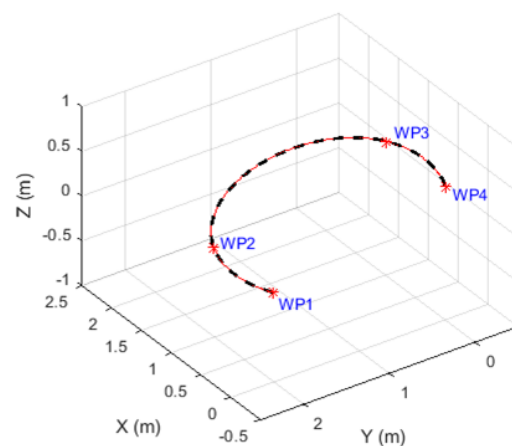


Figure 19. Simple 4-waypoint aggressive trajectory.

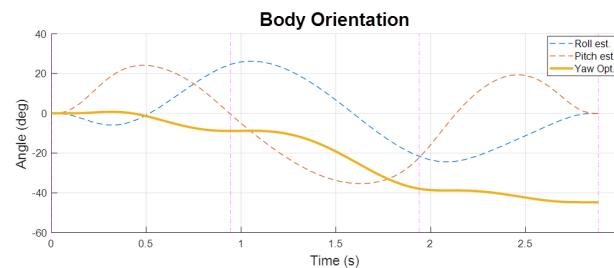


Figure 20. Optimal yaw trajectory for the trajectory.

We converted this optimal yaw trajectory into a polynomial form and used the geometric controller to track this aggressive polynomial trajectory. Figure 21 shows that, by tracking this optimal yaw trajectory, the M_z control input will be indeed close to zero.

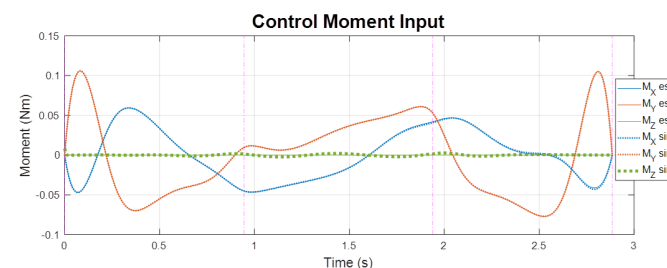


Figure 21. Simulation result of control input with geometric controller.

Figure 22 shows the comparison of rotor thrust used to track the aggressive trajectory between constant zero yaw and the optimal yaw trajectory. The additional yaw control effort M_z in the constant zero yaw case pulls the F_1 and F_3 rotor forces away from the F_2

and F_4 rotor forces and thereby increases the maximum rotor force needed. From the result, we can see that, by tracking the optimal yaw trajectory, the maximum rotor force is reduced significantly by 17.5% (from 4 N to 3.3 N).

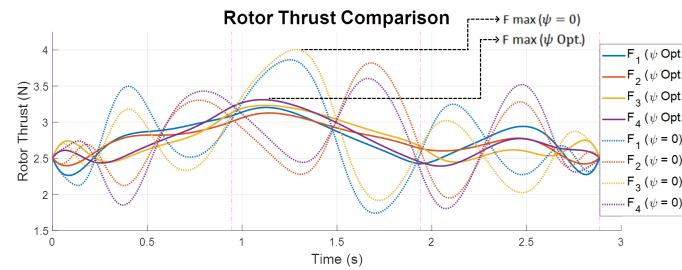


Figure 22. Rotor force comparison.

Remark 1. With the optimization and analysis framework developed, we review the choice of minimum snap trajectory. For the same four-waypoint scenario and segment times, we compare the optimal trajectory and the maximum rotor force required with the cost of different orders of trajectory derivatives to minimize. Figure 23 shows the geometry of the trajectories. As shown in Figure 24, the trajectory that minimizes the cost on the fourth derivative has the minimum force required. Therefore, we conclude that minimum snap is indeed the optimal choice for this approach of multi-copter aggressive trajectory optimization.

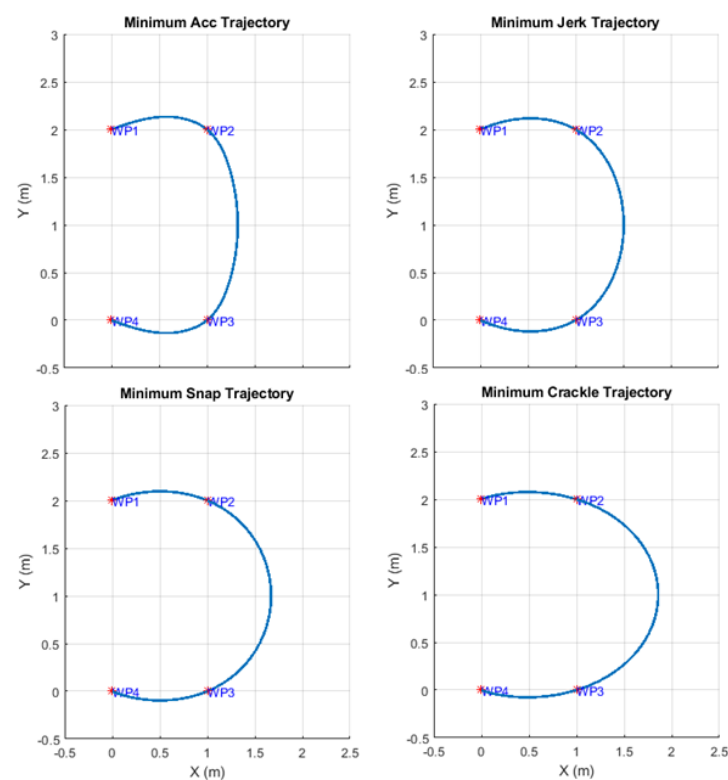


Figure 23. Comparison of trajectories optimized with cost on different order of derivative.

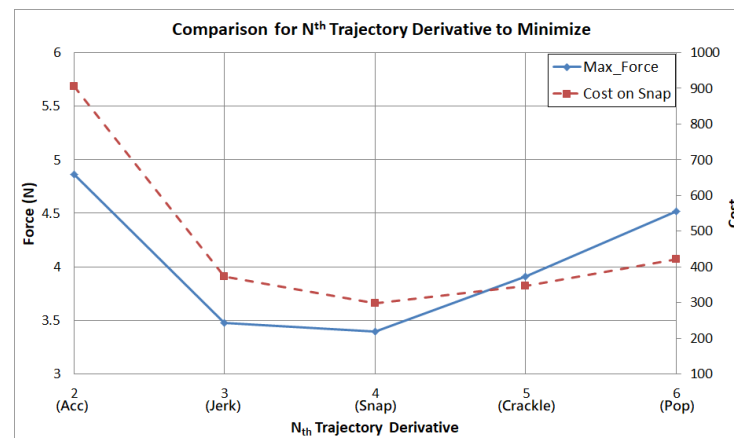


Figure 24. Maximum rotor force required for trajectories optimized with cost on different order of derivative.

8. Conclusions

With the framework developed in this research, we can find the optimal polynomial trajectory and the corresponding segment time allocation for a given quadcopter model, a specified scenario, and a desired aggressiveness. We can also tell if the given scenario is beyond the capability of the given quadcopter by examining whether the solution exists. Furthermore, we can use the algorithm to evaluate the minimum required rotor thrust to achieve the scenario. Instead of tracking the maximum force, we can also use the algorithm to track the total time. That is, given a desired flight time, we can find the optimal trajectory that has the minimum required rotor thrust. Though the quadcopter model is used in this paper, the inverse dynamics analysis for control input (collective force and three-axis moment) applies to generic multi-copters. Therefore, the method developed in this research can be applied to multi-copters with minor modifications to the vehicle control allocation model. The geometric controller used in this research showed the capability of tracking the optimized aggressive trajectory. The tracking result verified the feasibility of the optimized trajectory and the credibility of the inverse dynamics analysis. In addition, the yaw trajectory optimization method is capable of improving the aggressive performance in the case of no requirement on heading angle.

Author Contributions: Both K.S. and T.-L.L. were involved in the conceptualization and development of methodology. T.-L.L. performed the simulations and wrote the article. K.S. reviewed and edited the manuscript for publication. All analyses were jointly performed by T.-L.L. and K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Office of Naval Research under grant number N00014-18-1-2215 and The National Science Foundation (S&AS) under grant number 1724248. The APC was funded by Kamesh Subbarao.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledge the support from the Office of Naval Research under grant number N00014-18-1-2215 and the National Science Foundation (S&AS) under grant number 1724248.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tal, E.; Karaman, S. Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness. *IEEE Trans. Control. Syst. Technol.* **2020**, *29*, 1203–1218.
2. Gao, F.; Wu, W.; Pan, J.; Zhou, B.; Shen, S. Optimal Time Allocation for Quadrotor Trajectory Generation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018.
3. Yu, G.; Cabecinhas, D.; Cunha, R.; Silvestre, C. Quadrotor trajectory generation and tracking for aggressive maneuvers with attitude constraints. *IFAC-PapersOnLine* **2019**, *52*, 55–60.
4. Richter, C.; Bry, A.; Roy, N. Polynomial Trajectory Planning for Quadrotor Flight. In Proceedings of the International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013.
5. Mellinger, D.; Kumar, V. Minimum snap trajectory generation and control for quadrotors. In Proceedings of the International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011.
6. Mellinger, D.W. Trajectory Generation and Control for Quadrotors. *Publicly Access. Penn Diss.* **2012**, *547*, 5–9.
7. Richter, C.; Bry, A.; Roy, N. Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments. In *Robotics Research*; Inaba, M.; Corke, P., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 649–666.
8. Bertsekas, D.P. *Nonlinear Programming*; Athena Scientific: Belmont, MA, USA, 1999.
9. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, MA, USA, 2004.
10. Quan, Q. *Introduction to Multicopter Design and Control*; Springer: Singapore, 2017.
11. Lee, T.; Leok, M.; McClamroch, N. Geometric tracking control of a quadrotor uav on SE(3). In Proceedings of the 49th IEEE Conference on Decision and Control, Atlanta, GA, USA, 15–17 December 2010.