Quadrocopter Performance Benchmarking Using Optimal Control

Robin Ritz, Markus Hehn, Sergei Lupashin, and Raffaello D'Andrea

***Abstract*— A numerical method for computing quadrocopter maneuvers between two states is presented. Computed maneuvers satisfy Pontryagin's minimum principle with respect to time-optimality. First, in order to obtain the structure of time-optimal maneuvers, we apply the minimum principle to a first-principles, two-dimensional quadrotor model. Then we present a numerical algorithm that enables the computation of maneuvers for arbitrary initial and final states. The developed method is used to compute a set of maneuvers, which are discussed and demonstrated experimentally in the ETH Zurich Flying Machine Arena testbed.**

I. Introduction

The quadrocopter is a popular miniature aerial vehicle platform. A major reason for this popularity is the exceptional agility that these vehicles provide. For the rotational degrees of freedom, this is due to the off-center mounting of the propellers, combined with low rotational inertia. Translational dynamics are also typically fast, due to high thrust-to-weight ratios when not carrying a payload.

From a controls perspective, most early research on quadrotor dynamics focused on near-hover operation (see, for example, [1], and references therein). More dynamic maneuvers have been performed in recent years, including fast translations [2] and flips [3]. Algorithms that generate trajectories from a class of motion primitives (lines, polynomials, or splines) and which respect the dynamic constraints of quadrocopters have been introduced by several authors [4]–[6]. Dynamic feasibility is enforced by adapting the trajectory speed such that feasibility constraints are not violated.

A more specific trajectory generation problem is the generation of time-optimal trajectories between two states. A method using nonlinear programming and genetic algorithms has been presented [7]. It numerically minimizes the terminal time of the transition between two states, assuming piecewise constant control inputs.

In this paper, we present an algorithm that calculates control input and state trajectories between two states. We seek to find time-optimal trajectories and leverage Pontryagin's minimum principle to determine the structure of such trajectories. The calculated trajectories fulfill the minimum principle, making them strong candidates for optimality. Fig. 1 shows sample maneuvers that have been computed using the algorithm introduced in this paper.

The calculation of time-optimal maneuvers offers insight into the dynamical capabilities of quadrotor vehicles. While this algorithm is too slow to be used in real-time trajectory generation settings, it offers a valuable reference to benchmark other trajectory generation tools and controllers.

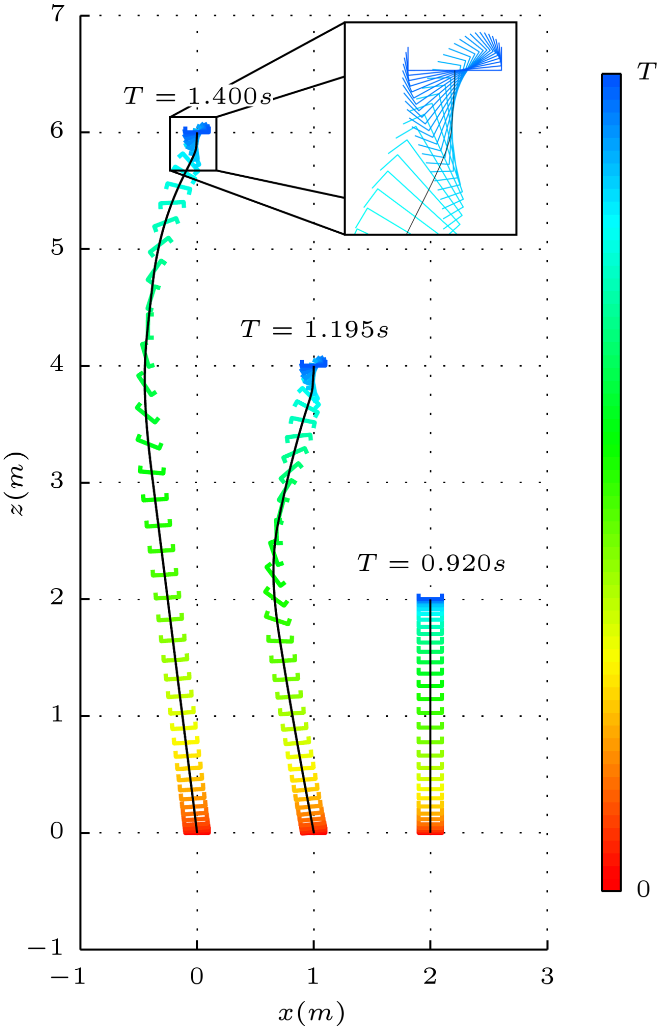

Fig. 1. Illustration of maneuvers for a purely vertical displacement of 2 m, 4 m, and 6 m. The maneuvers satisfy the minimum principle and for each maneuver, a quadrocopter is plotted every 0.03 s or every 0.01 s in the zoom box, respectively. The model parameters are based on the ETH Zurich Flying Machine Arena vehicles.

The remainder of this paper is structured as follows: In Section II, the first-principles dynamic model of the quadrocopter is presented. In Section III, the structure of time-optimal quadrotor control trajectories is derived. In Section IV, we present the algorithms used to compute solutions to the minimum principle for given initial and final states. Section V presents resulting maneuvers for a selection of translations. Section VI presents experimental results

The authors are with the Institute for Dynamic Systems and Control, ETH Zürich
{rritz, hehnm, sergeil, rdandrea}@ethz.ch

demonstrating the validity of the calculated trajectories, and Section VII provides a conclusion.

II. Quadrocopter Model

In this section, a two-dimensional model of the quadrocopter is presented and a non-dimensionalizing coordinate transformation is applied, which allows to describe the quadrocopter using only two parameters.

The two-dimensional quadrocopter model has three degrees of freedom: the horizontal position x, the vertical position z, and the pitch angle θ, as shown in Fig. 2. We assume that the angular velocity $\dot{\theta}$ can be set directly without dynamics and delay. This is motivated by the very high angular accelerations that quadrocopters can reach (typically on the order of several hundred rad/s^2), while the angular velocity is usually limited by the gyroscopic sensors used for feedback control on the vehicle [3]. As demonstrated by experimental results shown in Section VI, the model mismatch caused by this assumption is small, and it leads to tractable derivations in the following chapters.

The quadrocopter is controlled by two inputs: the total thrust force F_T and the pitch rate ω. Both controls are subject to saturation:

$$\underline{F_T} \leq F_T \leq \overline{F_T}, \quad |\omega| \leq \overline{\omega}. \tag{1}$$

Because commonly available motor drivers do not allow the direction of rotation to reverse mid-flight, and because the propellers are of fixed-pitch type, we assume that the thrust is always positive, i.e. $\underline{F_T} > 0$. The equations of motion are

$$\ddot{x} = \frac{F_T}{m}\sin\theta, \ \ddot{z} = \frac{F_T}{m}\cos\theta - g, \ \dot{\theta} = \omega, \tag{2}$$

where g denotes the gravitational acceleration and m the mass of the quadrocopter.

To describe the system using as few parameters as possible, we introduce the non-dimensionalizing coordinate transformation to unit gravity and unit allowable rotational rate:

$$\hat{t} = \overline{\omega} t, \ \hat{x} = \overline{\omega}^2 x/g, \ \hat{z} = \overline{\omega}^2 z/g. \tag{3}$$

This transformation is applied to (2). We define the state vector to be $\mathbf{x} = (\hat{x}, \dot{\hat{x}}, \hat{z}, \dot{\hat{z}}, \theta)$, and the control vector as $\mathbf{u} = (u_R, u_T) = (\omega/\overline{\omega}, F_T/(mg))$. The dimensionless quadrotor dynamics yield

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\hat{x}} \\ \ddot{\hat{x}} \\ \dot{\hat{z}} \\ \ddot{\hat{z}} \\ \dot{\theta} \end{pmatrix} = f(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} \dot{\hat{x}} \\ u_T \sin\theta \\ \dot{\hat{z}} \\ u_T \cos\theta - 1 \\ u_R \end{pmatrix}. \tag{4}$$

Note that the time derivative $\dot{\mathbf{x}}$ refers to the dimensionless case $\partial \mathbf{x}/\partial \hat{t}$. The transformation of the control inputs results in

$$\underline{U_T} \leq u_T \leq \overline{U_T}, \quad |u_R| \leq 1, \tag{5}$$

with $\underline{U_T} = \underline{F_T}/(mg)$ and $\overline{U_T} = \overline{F_T}/(mg)$. The transformed system dynamics are characterized by only two parameters, $\underline{U_T}$ and $\overline{U_T}$. In the following, the hat notation is dropped to increase readability. It is understood that all calculations are done in the dimensionless coordinate system.

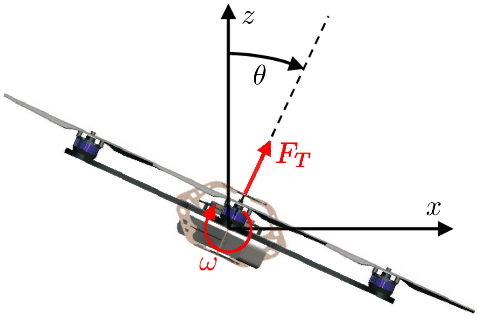

Fig. 2. Coordinate system and control inputs of the two-dimensional quadrocopter model.

III. Minimum Principle for Time-Optimal Quadrocopter Control

This section shows how Pontryagin's minimum principle is applied to time-optimal quadrocopter control. It follows that the thrust input is bang-bang, and the rotational control is bang-singular, meaning that the control input is always at full positive or negative saturation, except during singular arcs.

We seek to compute time-optimal maneuvers that bring the quadrocopter from a given initial state $\mathbf{x}_0$ to a given final state $\mathbf{x}_T$. An optimal maneuver is characterized by its state trajectory $\mathbf{x}^*(t), t \in [0, T]$, or the corresponding control inputs $\mathbf{u}^*(t), t \in [0, T]$. The time-optimal maneuver for given $\mathbf{x}_0$ and $\mathbf{x}_T$ is the solution to optimization problem

$$\begin{array}{ll} \text{minimize} & T \\ \text{subject to} & \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \\ & \mathbf{x}(0) = \mathbf{x}_0, \\ & \mathbf{x}(T) = \mathbf{x}_T, \\ & \mathbf{u} \in \mathbf{U} \ \forall t \in [0, T], \end{array} \tag{6}$$

where $\mathbf{U}$ denotes the set of all attainable control vectors. This problem can be solved using Pontryagin's minimum principle, which provides necessary conditions for optimality [8], [9]. With a cost function equal to 1, i.e. $g(\mathbf{x}, \mathbf{u}) = 1$, the Hamiltonian for the problem yields

$$\begin{aligned} H(\mathbf{x}, \mathbf{u}, \mathbf{p}) &= g(\mathbf{x}, \mathbf{u}) + \mathbf{p}^T f(\mathbf{x}, \mathbf{u}) \\ &= 1 + p_1 \dot{x} + p_2 u_T \sin\theta \\ &\quad + p_3 \dot{z} + p_4 (u_T \cos\theta - 1) + p_5 u_R \\ &= 0, \end{aligned} \tag{7}$$

where p_i denotes the elements of the costate vector $\mathbf{p}$. Note that along the optimal solution the Hamiltonian is zero for all times, i.e. $H \equiv 0$, since the terminal time T is free [8]. Applying the adjoint equation

$$\dot{\mathbf{p}} = -\nabla_{\mathbf{x}} H(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p}), \tag{8}$$

the first four costates result in

$$\begin{array}{ll} \dot{p}_1 = 0, & p_1 = c_1, \\ \dot{p}_2 = -p_1, & p_2 = c_2 - c_1 t, \\ \dot{p}_3 = 0, & p_3 = c_3, \\ \dot{p}_4 = -p_3, & p_4 = c_4 - c_3 t, \end{array} \tag{9}$$

with the unknown constants $\mathbf{c} = (c_1, c_2, c_3, c_4)$. For the remaining element p_5, the adjoint equation (8) gives

$$\dot{p}_5 = -p_2 u_T^* \cos\theta^* + p_4 u_T^* \sin\theta^*, \tag{10}$$

and p_5 cannot be expressed explicitly because it depends on the control input u_T^*, the trajectory of which is not known a priori.

A. *Optimal Control Inputs*

The minimum principle states that the optimal control input trajectory minimizes the Hamiltonian (7) over all possible values of $\mathbf{u}$. Since the two control inputs do not appear in the same summand, the Hamiltonian can be minimized separately for u_R and u_T:

1) Optimal Control Input u_R^*: For the rotational control input u_R, minimizing (7) results in

$$u_R^* = \underset{u_R \in [-1,+1]}{\operatorname{argmin}} \{p_5 u_R\}. \tag{11}$$

If p_5 changes sign, then u_R^* switches from -1 to $+1$ or vice versa. We define

$$\Phi_R = p_5 \tag{12}$$

as the switching function of u_R^*. If Φ_R is zero for a nontrivial interval of time, then the minimum condition (11) is insufficient to determine u_R^*. In these intervals, which are called singular arcs, u_R^* is determined using the condition that Φ_R remains zero. It follows that $\dot{\Phi}_R$ vanishes, which results in the condition

$$\dot{\Phi}_R = -p_2 u_T^* \cos\theta^* + p_4 u_T^* \sin\theta^* = 0. \tag{13}$$

Solving for θ^* using $u_T^* > 0$ yields

$$\theta^* = \arctan\left(\frac{p_2}{p_4}\right) = \arctan\left(\frac{c_2 - c_1 t}{c_4 - c_3 t}\right). \tag{14}$$

Differentiating (14) with respect to time gives the trajectory of the control input $u_R^* = \dot{\theta}^*$ in a singular arc:

$$u_{R,sing}^* = \frac{c_2 c_3 - c_1 c_4}{(c_1^2 + c_3^2)t^2 - 2(c_1 c_2 + c_3 c_4)t + c_2^2 + c_4^2}. \tag{15}$$

The rotational control input u_R^* of a time-optimal maneuver of the quadrocopter can be written as

$$u_R^* = \begin{cases} +1 & \text{if } \Phi_R < 0 \\ u_{R,sing}^* & \text{if } \Phi_R = 0 \\ -1 & \text{if } \Phi_R > 0 \end{cases}. \tag{16}$$

This type of optimal control trajectory is referred to as bang-bang singular control or bang-singular control [10], [11].

2) Optimal Control Input u_T^*: To compute the optimal control trajectory for the thrust input u_T^*, the sum of all terms of the Hamiltonian containing u_T must be minimized:

$$u_T^* = \underset{u_T \in [\underline{U_T}, \overline{U_T}]}{\operatorname{argmin}} \{p_2 u_T \sin\theta^* + p_4 u_T \cos\theta^*\}. \tag{17}$$

Again, we define a switching function

$$\Phi_T = p_2 \sin\theta^* + p_4 \cos\theta^*. \tag{18}$$

For a singular arc to exist, Φ_T must be zero for a nontrivial interval of time. Setting Φ_T to zero and solving for θ^* yields

$$\theta^* = \arctan\left(-\frac{p_4}{p_2}\right) = \arctan\left(\frac{c_3 t - c_4}{c_2 - c_1 t}\right). \tag{19}$$

The pitch angle θ^* is determined by the rotational control input u_R^*. It can be shown that (19) never holds for a nontrivial interval of time, neither in a singular, nor in a regular interval of u_R^*. The input u_T^* therefore does not contain singular arcs and can be written as

$$u_T^* = \begin{cases} \overline{U_T} & \text{if } \Phi_T \leq 0 \\ \underline{U_T} & \text{if } \Phi_T > 0 \end{cases}. \tag{20}$$

B. *Augmented System*

As only the derivative $\dot{\Phi}_R$ is given, we augment the system equations (4) with an additional state in order to represent the switching function Φ_R. We define $\mathbf{x_a} = (\mathbf{x}^*, \Phi_R)$, resulting in the augmented system dynamics

$$f_a(t, \mathbf{x_a}) = \begin{pmatrix} \dot{x}^* \\ u_T^* \sin\theta^* \\ \dot{z}^* \\ u_T^* \cos\theta^* - 1 \\ u_R^* \\ (c_1 t - c_2) u_T^* \cos\theta^* + (c_4 - c_3 t) u_T^* \sin\theta^* \end{pmatrix}. \tag{21}$$

The control inputs u_R^* and u_T^* are given by the control laws (16) and (20). A quadrocopter maneuver from $\mathbf{x}_0$ to $\mathbf{x}_T$ that satisfies the minimum principle solves the boundary value problem (BVP)

$$\begin{aligned} \dot{\mathbf{x}}_\mathbf{a} &= f_a(t, \mathbf{x_a}), \\ \mathbf{x}^*(0) &= \mathbf{x}_0, \\ \mathbf{x}^*(T) &= \mathbf{x}_T. \end{aligned} \tag{22}$$

In order to solve BVP (22), the final time T and the unknown constants $\mathbf{c}$ must be determined.

This completes the derivation of the minimum principle for time-optimal quadrocopter control given the model presented in Section II. The rotational control input u_R^* is bang-singular with switching function (12) and singular input (15), and the thrust input u_T^* is bang-bang with switching function (18). Finally, the augmented system dynamics are given by (21).

IV. Computation of Time-Optimal Maneuvers

In this section, a numerical method for computing quadrocopter maneuvers between two arbitrary states is introduced. The resulting maneuvers satisfy the minimum principle with respect to time-optimality.

To solve BVP (22), we use the approach of switching time optimization (STO), as proposed in [12]. A significant advantage of STO for this problem is that, if the maneuver is bang-bang without singular arcs, it does not require an initial guess of the unknown constants $\mathbf{c}$.

A. Bang-Bang Maneuvers

First, maneuvers with pure bang-bang behavior of both control inputs are considered, meaning that it is assumed that no singular arcs occur in u_R^*. The algorithm consists of three successive steps: 1) a bang-bang maneuver is found that brings the quadrocopter from the initial to the desired final state using STO: The maneuver is obtained by varying the duration T and the times at which the control inputs switch between their boundaries, until the quadrocopter reaches the final state with an acceptable accuracy; 2) based on the resulting maneuver trajectory, conditions on the constant vector $\mathbf{c} = (c_1, c_2, c_3, c_4)$ for BVP (22) to be satisfied by the maneuver are set up. These conditions are used to extract a first guess of the unknown constants $\mathbf{c}$; 3) having a good initial guess of the maneuver duration T and of the constant vector $\mathbf{c}$, a BVP solver that computes a solution to BVP (22) is applied. In the following, these three steps are introduced in detail:

1) Switching Time Optimization: Due to the assumption that the optimal solution is a bang-bang maneuver, the control trajectory $\mathbf{u}(t)$ can be efficiently parameterized by the initial control inputs $\mathbf{u}(t = 0)$ and the switching times of the two control inputs, denoted by the two sets

$$\begin{aligned} \{T_{u_R}\} &= T^i_{u_R} \quad \text{for } i = 1, 2, \ldots, N_R, \\ \{T_{u_T}\} &= T^j_{u_T} \quad \text{for } j = 1, 2, \ldots, N_T. \end{aligned} \tag{23}$$

N_R and N_T are the number of switches of the corresponding control input. The principle of STO is to choose N_R and N_T and to then improve an initial choice of the switching times, until a control trajectory is found that guides the quadrocopter from $\mathbf{x}_0$ to $\mathbf{x}_T$ with an acceptable accuracy. We measure the final state error using the scalar final state residual function

$$P_{res}(\{T_{u_R}\}, \{T_{u_T}\}, T) = (\mathbf{x}(T) - \mathbf{x}_T)^T W (\mathbf{x}(T) - \mathbf{x}_T), \tag{24}$$

where the matrix $W = \text{diag}(w_1, w_2, w_3, w_4, w_5)$ contains the weights of the different state errors. The final state $\mathbf{x}(T)$ resulting from the chosen switching times can be obtained by numerically integrating the system dynamics $f(\mathbf{x}, \mathbf{u})$ over the interval $[0, T]$, where $\mathbf{u}$ is defined by the initial control inputs $\mathbf{u}(t = 0)$ and the switching times $\{T_{u_R}\}$ and $\{T_{u_T}\}$. Note that the maneuver duration T is not known a priori and we must find the minimum T for which $P_{res} = 0$ can be obtained. The problem can be written as

$$\begin{aligned} &\text{find} && \{T_{u_R}\}, \{T_{u_T}\}, T \\ &\text{subject to} && P_{res}(\{T_{u_R}\}, \{T_{u_T}\}, T) = 0, \\ &&& T \leq \{T\}_{ach}, \end{aligned} \tag{25}$$

where $\{T\}_{ach}$ is the set of all T for which $P_{res} = 0$ is achievable.

The solution of (25) is found by a two-step algorithm: For an initially small, fixed maneuver duration T, the state residual P_{res} is minimized by varying the switching times $\{T_{u_R}\}$ and $\{T_{u_T}\}$. After the minimization, T is increased using the secant method

$$T_{i+1} = T_i + \frac{T_i - T_{i-1}}{(P_{res,i-1}/P_{res,i}) - 1}, \tag{26}$$

or by a constant value if the secant method does not converge [13]. These two steps are repeated until $P_{res} = 0$ is achieved. Since the initial value of T is chosen to be too small to complete the maneuver, and since T is successively increased, the algorithm delivers the smallest T for which $P_{res} = 0$ is achievable.

2) Parameter Extraction: After having found a bang-bang trajectory that brings the quadrocopter from the initial state $\mathbf{x}_0$ to the desired final state $\mathbf{x}_T$, it is necessary to verify that it is a solution to BVP (22). Therefore, the constant vector $\mathbf{c} = (c_1, c_2, c_3, c_4)$ must be determined, based on the trajectories resulting from the STO. Several conditions on $\mathbf{c}$ can be constructed from the control input trajectories:

a) Conditions on the Trajectory of Φ_R: The switching function Φ_R has a zero-crossing when the control input u_R switches, which leads to the conditions

$$\Phi_R(T^i_{u_R}) = 0 \quad \text{for } i = 1, 2, \ldots, N_R. \tag{27}$$

As shown in Section III, only the derivative $\dot{\Phi}_R$ of the switching function is known a priori. However, once the state trajectories are known from the STO, the condition $H \equiv 0$ can be used to compute Φ_R.

The derivative $\dot{\Phi}_R$ is given by (21). For a trajectory that satisfies the minimum principle, the integral of $\dot{\Phi}_R$ must coincide with the trajectory of Φ_R. Hence, for an arbitrary interval $[t_1, t_2] \in [0, T]$,

$$\Phi_R(t_2) - \Phi_R(t_1) = \int_{t_1}^{t_2} \dot{\Phi}_R dt \tag{28}$$

must hold, where the left side of the equation is computed using $H \equiv 0$. To define conditions on $\mathbf{c}$ based on (28), the maneuver interval $[0, T]$ is divided into $N_R + 1$ subintervals that are separated by the switching times $\{T_{u_R}\}$. This choice is beneficial with respect to the computational effort because the switching function Φ_R must vanish at the switching times; the left side of (28) can be set to zero for all intervals, except for the first and the last one.

b) Conditions on the Trajectory of Φ_T: Φ_T must vanish at each switch of u_T, hence

$$\Phi_T(T^i_{u_T}) = 0 \quad \text{for } i = 1, 2, \ldots, N_T, \tag{29}$$

where $\{T_{u_T}\}$ is given by the STO.

c) Condition Matrix Equation: For the minimum principle to be satisfied, a constant vector $\mathbf{c}$ that fulfills the above conditions to an acceptable accuracy must exist. It can be shown that all conditions are linear with respect to $\mathbf{c}$. Hence, the conditions from (27), (28), and (29) can be merged in a linear system of equations, denoted by the matrix equation

$$A\mathbf{c} = r. \tag{30}$$

Since the system is overdetermined for all maneuvers considered here, the least squares solution of (30) is computed [14], which is given by

$$\mathbf{c}^* = (A^T A)^{-1} A^T r. \tag{31}$$

Then, $\mathbf{c}^*$ is substituted back into (30) and the error is checked. If the error is not close to zero, then the minimum principle is not satisfied by the maneuver. Reasons for this can be a wrong number of switches, or the existence of singular arcs.

3) BVP Solver: To verify that BVP (22) is fulfilled and to minimize numerical errors, we perform a last step, where the BVP is solved numerically. The state residual P_{res} is minimized by varying the constant vector $\mathbf{c}$ and the maneuver duration T. The problem can be written as

$$\begin{array}{ll} \text{minimize} & P_{res}(\mathbf{c}, T) \\ \text{subject to} & \dot{\mathbf{x}}_\mathbf{a} = f_a(t, \mathbf{x}_\mathbf{a}), \\ & \mathbf{x}_\mathbf{a}(0) = (\mathbf{x}_0, \Phi_R(0)). \end{array} \tag{32}$$

The constants $\mathbf{c}$ resulting from the parameter extraction and the maneuver duration T obtained by the STO are used as initial values. As these initial values are close to the exact solution, the BVP solver converges quickly. $\Phi_R(0)$ can be obtained by the condition $H \equiv 0$ evaluated at $t = 0$. If P_{res} is sufficiently small after the minimization, the solution satisfies the minimum principle.

B. Bang-Singular Maneuvers

We now consider maneuvers where u_R contains singular arcs. The algorithm is similar to the one for bang-bang maneuvers introduced in Section IV-A. However, the steps have to be modified to take the singular arcs into account: The parameter extraction step is embedded into the STO, since during the singular arcs, the trajectory of u_R depends on the constants $\mathbf{c}$. The principle of STO with embedded parameter extraction is to find a maneuver that brings the quadrocopter to the desired final state, and in parallel, to find a constant vector $\mathbf{c}$ that fulfills all conditions resulting from the parameter extraction. Hence, for bang-singular maneuvers, the algorithm consists of two successive steps: 1) the STO with embedded parameter extraction; and 2) the BVP solver.

1) Switching Time Optimization with Embedded Parameter Extraction: The approach of STO is again used to find a control trajectory that brings the quadrocopter from $\mathbf{x}_0$ to $\mathbf{x}_T$. However, for bang-singular maneuvers, u_R can stay within a singular arc for a particular duration, each time it switches. A new set of parameters is introduced that describes the durations of the singular arcs: We denote the duration within the singular arc at the switching time $T^i_{u_R}$ as D^i_{s,u_R}: At the time $T^i_{u_R}$ the control input u_R enters the singular arc, and at time $T^i_{u_R} + D^i_{s,u_R}$ the singular arc is left and u_R switches to -1 or $+1$. Thus, the maneuver is characterized by the sets $\{T_{u_R}\}$, $\{T_{u_T}\}$, and $\{D_{s,u_R}\}$. Within a singular arc, u_R is given by (15) and its trajectory depends on the constants $\mathbf{c}$. The final state residual P_{res} is therefore also a function of $\mathbf{c}$ and can be written as

$$\begin{aligned} P_{res}(\{T_{u_R}\}, \{T_{u_T}\}, \{D_{s,u_R}\}, \mathbf{c}, T) \\ = (\mathbf{x}(T) - \mathbf{x}_T)^T W (\mathbf{x}(T) - \mathbf{x}_T). \end{aligned} \tag{33}$$

$\{D_{s,u_R}\}$ and $\mathbf{c}$ are additional optimizing variables during the STO. Since the dynamics are extremely sensitive to $\mathbf{c}$, a good initial guess is required. The constant vector $\mathbf{c}$ from a maneuver with similar $\mathbf{x}_0$ and $\mathbf{x}_T$ is therefore taken as initial value. Since only constants $\mathbf{c}$ that fulfill the linear matrix equation $A\mathbf{c} = r$ from the parameter extraction are a valid choice, we define the residual of the matrix condition equation as

$$C_{res}(\{T_{u_R}\}, \{T_{u_T}\}, \{D_{s,u_R}\}, \mathbf{c}, T) = (A\mathbf{c}-r)^T W_c (A\mathbf{c}-r), \tag{34}$$

where W_c is a diagonal matrix containing the weights of the different linear conditions. For a maneuver that satisfies the minimum principle, the condition residual C_{res} must vanish. Thus, the STO problem can be written as

$$\begin{array}{ll} \text{find} & \{T_{u_R}\}, \{T_{u_T}\}, \{D_{s,u_R}\}, \mathbf{c}, T \\ \text{subject to} & P_{res}(\{T_{u_R}\}, \{T_{u_T}\}, \{D_{s,u_R}\}, \mathbf{c}, T) = 0, \\ & C_{res}(\{T_{u_R}\}, \{T_{u_T}\}, \{D_{s,u_R}\}, \mathbf{c}, T) = 0, \\ & T \leq \{T\}_{ach}, \end{array} \tag{35}$$

where $\{T\}_{ach}$ denotes the set of all T for which $P_{res} = 0$ and $C_{res} = 0$ is achievable.

For bang-singular maneuvers, the sum of the state and the condition residual $P_{res} + C_{res}$ is minimized during the STO. For the computation of C_{res}, the matrix A and the vector r are required. This means that the parameter extraction (which is the second step in the bang-bang case and delivers A and r) is no longer an isolated step, but needs to be called for each evaluation of C_{res} within the STO minimization.

Parameter Extraction for Bang-Singular Maneuvers: For the parameter extraction of bang-singular maneuvers, which is needed to obtain A and r, there exist additional linear conditions which take the requirements on the switching functions within singular arcs into account:

a) Conditions on the Trajectory of Φ_R: Within a singular arc, Φ_R must stay at zero, and to take this into account, we demand that Φ_R is zero at the beginning and at the end of the singular arcs. This leads to the conditions

$$\begin{array}{ll} \Phi_R(T^i_{u_R}) = 0 & \text{for } i = 1, 2, \dots, N_R, \\ \Phi_R(T^i_{u_R} + D^i_{s,u_R}) = 0 & \text{for } i = 1, 2, \dots, N_R. \end{array} \tag{36}$$

Note that these conditions do not imply that Φ_R is zero during the entire singular arc. It is therefore necessary to verify the trajectory of Φ_R after the STO.

As the derivative $\dot{\Phi}_R$ is known explicitly, for bang-bang maneuvers, we demanded that the integration value of $\dot{\Phi}_R$ between two switches is zero. For bang-singular maneuvers, we pose similar conditions, but extra time intervals over the singular arcs are created, to penalize constant drifts of Φ_R.

Because Φ_R stays at zero during the singular arcs, its derivative $\dot{\Phi}_R$ must vanish as well. Assuming that the thrust input u_T does not switch at the edges of the singular intervals[1], $\dot{\Phi}_R$ is continuous over the border of the singular arcs, as can be seen from (21). Consequently, the switching function Φ_R enters and leaves a singular arc tangentially. We therefore impose the conditions that the derivative $\dot{\Phi}_R$ is zero at the edges of every singular arc. For each singular

[1] The assumption that u_T does not switch at the edges of the singular arcs has not been proven. However, for all maneuvers considered here, this condition has been fulfilled.

arc, i.e. for each $D^i_{s,u_R} > 0$, two additional linear conditions result:

$$\begin{aligned} \dot{\Phi}_R(T^i_{u_R}) &= 0, \\ \dot{\Phi}_R(T^i_{u_R} + D^i_{s,u_R}) &= 0 \end{aligned} \tag{37}$$

b) Conditions on the Trajectory of Φ_T: The conditions on Φ_T are the same as for bang-bang maneuvers, because u_T does not contain singular arcs.

c) Condition Matrix Equation: As in the previous case, it can be shown that all conditions are linear with respect to the constant vector $\mathbf{c}$. Again, the linear conditions are combined into a matrix equation to obtain A and r.

2) BVP Solver: Similar to bang-bang maneuvers, the final step of the algorithm is reducing the errors by applying a BVP solver. If the maneuver contains singular arcs, Φ_R stays at zero for a nontrivial interval of time. Since the system is integrated numerically, Φ_R is near zero during the singular arcs, but does not vanish completely due to numerical inaccuracies. Since Φ_R enters and leaves the singular arcs tangentially, defining a threshold value below which Φ_R is considered to be zero is not a straightforward task. For this reason, the control trajectory $\mathbf{u}(t)$ is not determined using the optimal control laws (i.e. based on the switching functions), but based on the sets $\{T_{u_R}\}$, $\{T_{u_T}\}$, and $\{D_{s,u_R}\}$. To verify that the control inputs correspond to the optimal control laws, we plot the switching functions Φ_R and Φ_T after executing the BVP solver, and check if the control laws are satisfied. For bang-singular maneuvers, the BVP solver is very similar to the STO. The only (but important) difference is that the maneuver duration T is an optimization variable, too, and not kept constant during the minimization of $P_{res} + C_{res}$.

V. Results

In this section, resulting quadrocopter maneuvers that were computed using the algorithms introduced in Section IV are presented. The focus lies on maneuvers that lead to a horizontal and vertical displacement while the quadrocopter is at rest at the beginning and at the end of the maneuver.

Table I shows the used numerical model parameters in the dimensional form ($\underline{F_T}/m$, $\overline{F_T}/m$, $\overline{\omega}$). The non-dimensional parameters ($\underline{U_T}$, $\overline{U_T}$) can easily be calculated from these using the control input transformation (5). The model parameters are based on the ETH Zurich Flying Machine Arena vehicles [3]. For the visualization of the computed maneuvers, the system is transformed back to the state variables representing physical dimensions, which allows a more intuitive interpretation. Nevertheless, all results presented here can easily be transformed back to the more general non-dimensional case.

TABLE I
Numerical parameters of the quadrocopter model.

Parameter	Value	Description
$\underline{F_T}/m$	$1\,\mathrm{m/s^2}$	Minimum mass-normalized thrust
$\overline{F_T}/m$	$20\,\mathrm{m/s^2}$	Maximum mass-normalized thrust
$\overline{\omega}$	$10\,\mathrm{rad/s}$	Maximum rotational rate

A. Horizontal Displacements

First we consider maneuvers with purely horizontal displacements. We define the quadrocopter to be at rest and at a pitch angle of zero at the beginning of the maneuver. Without loss of generality, we assume that the initial position of the quadrocopter is at the origin:

$$\mathbf{x}_0 = (x(0), \dot{x}(0), z(0), \dot{z}(0), \theta(0)) = (0, 0, 0, 0, 0). \tag{38}$$

For purely horizontal displacements, we demand that, at the end of the maneuver, the quadrocopter is at rest again, has zero pitch, and no overall vertical displacement:

$$\mathbf{x}_T = (x(T), \dot{x}(T), z(T), \dot{z}(T), \theta(T)) = (x_T, 0, 0, 0, 0). \tag{39}$$

We assume that x_T is positive without loss of generality. Maneuvers have been computed for a horizontal displacement x_T from $0.1\,\mathrm{m}$ to $15\,\mathrm{m}$, with a step size of $0.1\,\mathrm{m}$. In Fig. 3, the maneuver duration T is plotted as a function of the horizontal displacement x_T. Furthermore, Fig. 3 shows the switching times for each maneuver. For a particular displacement x_T, the maneuver starts at $t = 0\,\mathrm{s}$, and as time passes, it moves in the positive direction of the t-axis. Every time a switching line is crossed, the corresponding control input switches to the value specified in the diagram.

For $x_T \leq 1.5\,\mathrm{m}$, the maneuver is bang-bang with no singular arcs. At the beginning the quadrocopter turns at maximum rate, and around the maximum pitch angle θ the thrust is switched on for acceleration. Then it turns in the negative direction to decelerate around the minimum peak of θ, before it goes back to $\theta = 0$. At $x_T = 1.6\,\mathrm{m}$, two singular arcs appear. Roughly speaking, the pitch angle is kept at $\theta \approx \pm\pi/2$ for acceleration and braking, respectively. Because a trade-off between fast acceleration in x and maintaining altitude in z is necessary, the pitch angle is

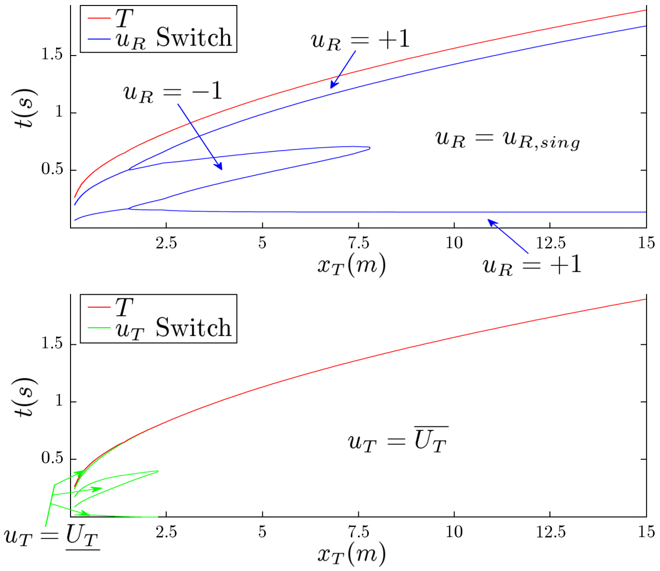

Fig. 3. Maneuver duration T as function of the final displacement x_T for purely horizontal maneuvers. Additionally, the switching times of u_R are drawn in the plot on the top, and the switching times of u_T in the plot on the bottom.

not exactly $\theta = \pm\pi/2$ within the singular arcs, and not constant. For $x_T \geq 7.9\,\mathrm{m}$, the two singular arcs merge: The quadrocopter turns smoothly to a negative θ for deceleration, instead of a sharp turn in the middle of the maneuver. Note also that for $x_T \geq 2.4\,\mathrm{m}$ the thrust is always at its maximum. Fig. 4 shows an illustration of some selected maneuvers.

B. Vertical Displacements

We now consider maneuvers for a purely vertical displacement. The initial and final state are:

$$\mathbf{x}_0 = (0,0,0,0,0), \quad \mathbf{x}_T = (0,0,z_T,0,0) \qquad (40)$$

Maneuvers have been computed for a displacement z_T ranging from $0.1\,\mathrm{m}$ to $10\,\mathrm{m}$, with a step size of $0.1\,\mathrm{m}$. Fig. 5 shows the maneuver duration T and the switching times as a function of z_T.

If the desired vertical displacement is small, i.e. for $z_T \leq 2.4\,\mathrm{m}$, the quadrocopter is within a singular arc during the entire maneuver. The pitch angle remains at exactly $\theta = 0$. The thrust is switched on at the beginning and switched off at a time such that the quadrocopter comes to rest due to gravity at the desired height z_T. For $z_T \geq 2.5\,\mathrm{m}$, it is beneficial to perform a flip and to make use of the thrust for braking while the pitch is around $\theta \approx \pm\pi$. For $z_T \geq 6.3\,\mathrm{m}$, a singular arc which keeps the pitch near $\theta \approx \pm\pi$ for a particular time appears, as can be seen in Fig. 5. Thus, the flip is stopped for an interval of deceleration. Some selected maneuvers are shown in Fig. 1.

C. General Displacements

The algorithm introduced in this paper has also been used to compute maneuvers which lead not only to a displacement in one direction, but to a general two-dimensional displacement. The initial and final state are:

$$\mathbf{x}_0 = (0,0,0,0,0), \quad \mathbf{x}_T = (x_T,0,z_T,0,0) \qquad (41)$$

Since the final state of a general displacement contains two variables (x_T and z_T), the maneuver duration T cannot be easily plotted in a two-dimensional figure. Instead, as an example, a maneuver with a displacement of $5\,\mathrm{m}$ in horizontal and vertical direction, i.e. a maneuver with $x_T = 5\,\mathrm{m}$ and $z_T = 5\,\mathrm{m}$, is illustrated here. Fig. 6 shows the resulting input, state, and switching function trajectories of this example maneuver. Note that the control inputs and the switching functions indeed fulfill the control laws (16) and (20). This implies that the minimum principle for time-optimality is satisfied.

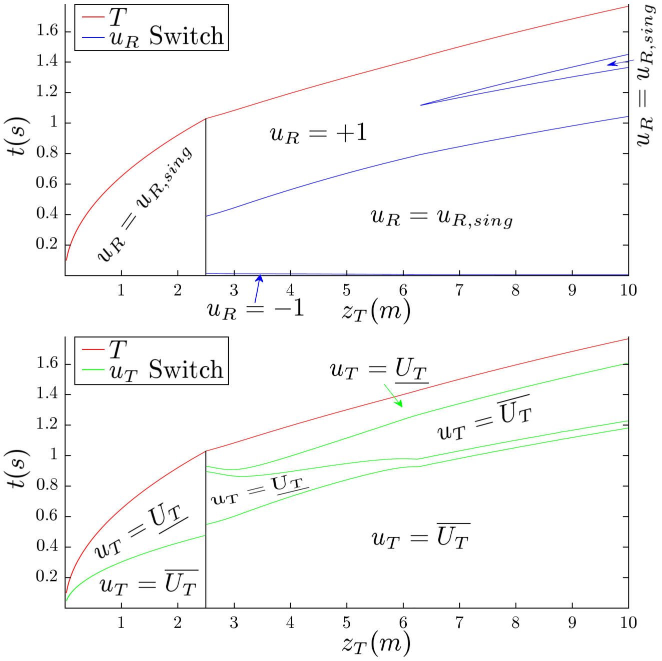

Fig. 5. Maneuver duration T as function of the final vertical displacement z_T for maneuvers with no horizontal displacement. Additionally, the switching times of u_R are drawn in the plot on the top, and the switching times of u_T in the plot on the bottom. The vertical black line denotes where the structure of the minimum principle solution changes: on the left the solution without a flip, which is faster for small z_T, and on the right the solution where the quadrocopter performs a flip, which is faster for large z_T.

VI. Experimental Results

This section presents the experimental validation of the numerical algorithms and resulting maneuvers introduced in the previous chapters. We observe that the experimental trajectories are similar to the ones computed numerically.

The experiments were carried out on quadrotor vehicles in the ETH Zurich Flying Machine Arena. We use modified Ascending Technologies 'Hummingbird' quadrocopters [15]. The vehicles are equipped with custom electronics that allow the deployment of custom control algorithms. Trajectories

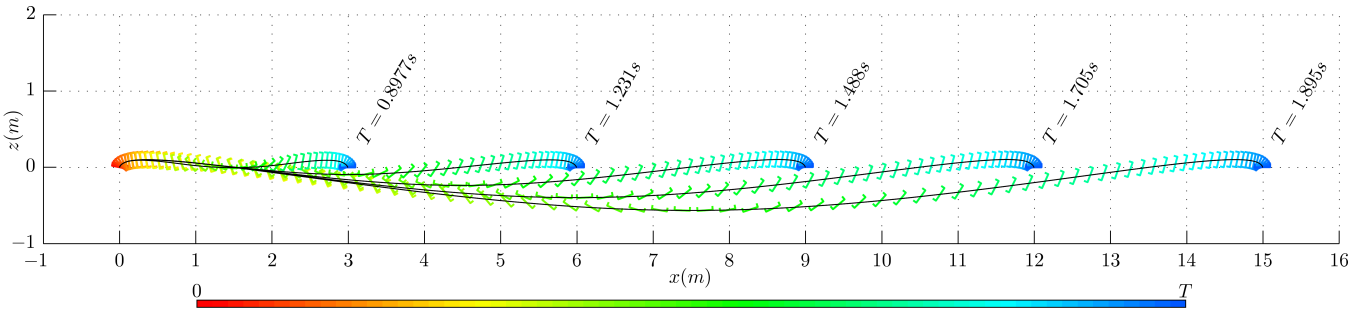

Fig. 4. Illustration of maneuvers for a purely horizontal displacement between 3 m and 15 m. The maneuvers satisfy the minimum principle and for each maneuver, a quadrocopter is plotted every 0.02 s.

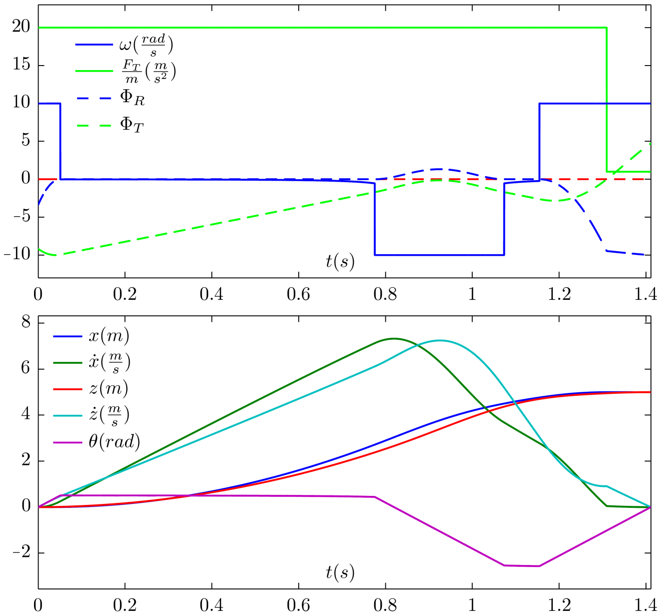

Fig. 6. Input, state, and switching function trajectories of an example maneuver with $x_T = 5\,\mathrm{m}$ and $z_T = 5\,\mathrm{m}$. In the plot of the control inputs, the switching functions are also drawn, but note that they are scaled to fit into the plot.

were recorded using an infrared motion tracking system. The control input trajectories are transfered to the quadrotor vehicle before the start of the experiment. When the vehicle is stabilized at the initial state, the maneuver is triggered and the vehicle executes the control input trajectories, using only feedback from the on board gyroscopes to control its rotational rates. The trajectory is sampled and executed at 800 Hz. In order to compensate for inaccuracies of the dynamical model used in this paper, a policy gradient learning method was applied to the maneuver [3]: The maneuver was started from the initial state, and the times of the control input switches were refined using a policy gradient method to achieve the desired terminal state.

Fig. 7 shows the state trajectories of a translation of 5 m in both coordinates after the learning algorithm was applied. This is the same maneuver as shown in Section V-C (Fig. 6). Note that the total maneuver duration is approximately 20% higher than the calculated duration (approximately 1.7 s versus 1.4 s). The pitch angle trajectory θ shows inaccuracies caused by unmodeled rotational accelerations, but the overall characteristics of the experimental trajectory are similar to the simulated trajectory. These results support the modeling presented in Chapter II, and demonstrate that the maneuvers calculated using the algorithm presented herein are realistic. A video showing this experiment is available online at `www.idsc.ethz.ch/people/staff/hehn-m`.

VII. Conclusion

A method that numerically computes quadrocopter maneuvers satisfying the minimum principle with respect to time-optimality has been developed. The algorithm is based on a first-principles dimensionless quadrocopter model, which is described by two parameters. Resulting maneuvers for some particular initial and final positions have been illustrated and discussed, and the transferability to real quadrocopters has been demonstrated in the ETH Zurich Flying Machine Arena testbed. We expect this method to be a valuable tool that allows performance benchmarking of other quadrocopter controllers, and that offers insights into the structure of time-optimal quadrocopter maneuvers.

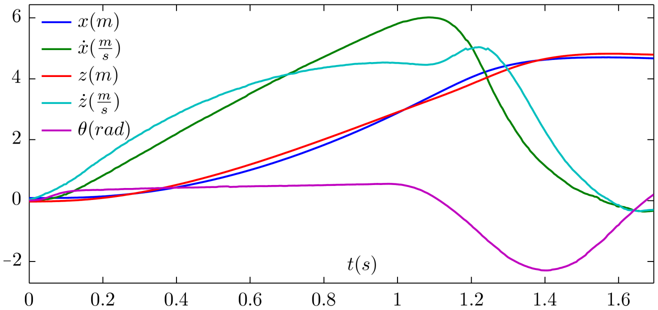

Fig. 7. Measured state trajectories of the example maneuver with $x_T = 5\,\mathrm{m}$ and $z_T = 5\,\mathrm{m}$, executed on the quadrocopter. The maneuver is identical to the one presented in Fig. 6. Note that the state trajectories on the real system are similar to the ones in simulation.

References

[1] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-Time Indoor Autonomous Vehicle Test Environment," *IEEE Control Systems Magazine*, April 2008.

[2] D. Mellinger, N. Michael, and V. Kumar, "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors," in *International Symposium on Experimental Robotics*, 2010.

[3] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A Simple Learning Strategy for High-Speed Quadrocopter Multi-Flips," in *IEEE International Conference on Robotics and Automation*, 2010.

[4] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Quadrotor Helicopter Trajectory Tracking Control," in *IEEE Conference on Decision and Control*, 2008.

[5] I. Cowling, O. Yakimenko, and J. Whidborne, "A Prototype of an Autonomous Controller for a Quadrotor UAV," in *European Control Conference*, 2007.

[6] Y. Bouktir, M. Haddad, and T. Chettibi, "Trajectory Planning for a Quadrotor Helicopter," in *Mediterranean Conference on Control and Automation*, 2008.

[7] L.-C. Lai, C.-C. Yang, and C.-J. Wu, "Time-Optimal Control of a Hovering Quad-Rotor Helicopter," *Journal of Intelligent and Robotic Systems*, vol. 45, no. 2, 2006.

[8] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Volume 1*. Athena Scientific, 2005.

[9] H. P. Geering, *Optimal Control with Engineering Applications*. Springer, 2007.

[10] L. Poggiolini and G. Stefani, "Minimum Time Optimality for a Bang-Singular Arc: Second Order Sufficient Conditions," in *IEEE Conference on Decision and Control*, 2005.

[11] S. A. Vakhrameev, "A Bang-Bang Theorem with a Finite Number of Switchings for Nonlinear Smooth Control Systems," *Journal of Mathematical Sciences*, 1997.

[12] M. J. Zandvliet, O. H. Bosgra, J. D. Jansen, P. M. J. V. den Hof, and J. F. B. M. Kraaijevanger, "Bang-Bang Control and Singular Arcs in Reservoir Flooding," *Journal of Petroleum Science and Engineering*, 2006.

[13] G. Dahlquist and Å. Björck, *Numerical Methods*. Courier Dover Publications, 1974.

[14] D. S. Bernstein, *Matrix Mathematics*. Princeton University Press, 2005.

[15] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz," in *IEEE International Conference on Robotics and Automation*, 2007.