International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, June 2016.

Analytical Time-optimal Trajectory Generation and Control for Multirotors

Marius Beul and Sven Behnke

***Abstract*— Micro aerial vehicles, such as multirotors, are particularly well suited for autonomous monitoring, inspection, and surveillance, e.g., for doing inventory in large warehouses.**

Key prerequisites for efficient MAV inventory missions is time- and energy-efficient flight. Generated trajectories must assure the visiting of certain waypoints while preventing large deviations or overshoot, as these can cause crashes in narrow corridors.

In this paper, we propose a novel trajectory generation method that is able to analytically compute time- and energy-optimal trajectories, incorporating system dynamics, based on a first-principles model. Due to the fast runtime of the method, it is not only used to generate trajectories on a higher level, but to control the MAV in real-time, substituting state-of-the-art cascaded control loops for position, velocity, and attitude.

I. Introduction

Micro aerial vehicles (MAVs) are becoming a key factor in reducing the required time, risks, and costs for, e.g., search and rescue missions, inspection tasks, and aerial photography. Due to their flexibility and low cost, they constitute a promising alternative to employing heavy machinery or even risking the health of humans in many situations. Especially for inventory in large warehouses, MAVs can save time and cost, replacing the cumbersome manual acquisition of the present stock by scanning on the fly.

However, to make such applications efficient, MAV flight needs to be autonomous and time-efficient. On one hand, time-efficient flight must be fast, so that more waypoints can be visited in a given time. On the other hand, time-efficient flight must also be energy efficient, so that the total flight time is maximized and flights back to the charging station are reduced to a minimum.

Since aisle widths of $3\,\mathrm{m}$ or less are common in warehouses, MAV flight also needs to be precise. Classic approaches, employing a cascaded control loop for position, velocity, and attitude are either conservatively slow or induce overshoot. Furthermore, these approaches are not capable to respect certain state constraints, e.g., maximum allowed velocity, and rely on a proper parameterization.

In this paper, we propose a novel trajectory generation method that generates time- and energy-optimal trajectories from the simplified dynamics of the MAV. Due to the fast runtime, our method is not only able to compute offline trajectories, but is also capable of controlling position,

This work has been supported by a grant of the German Bundesministerium für Wirtschaft und Energie in the Autonomics for Industry 4.0 project InventAIRy.

The authors are with the Autonomous Intelligent Systems Group, Computer Science VI, University of Bonn, Germany
mbeul@ais.uni-bonn.de

Fig. 1. Our MAV (circled) on an inventory mission. It is equipped with multiple cameras and an RFID reader to detect and locate inventory in a warehouse. Here, we scan and locate the righthand boxes in a prerecorded map of the warehouse.

velocity, and acceleration of the MAV in real-time. Since the method works parameterless, no cumbersome parameter identification is needed.

Optimality is guaranteed by exploiting specific properties of the environment:

- We assume that direct line of sight to the next target waypoint is available. When used in a hierarchy of planners, this constraint can always be met. In an orthogonal environment like warehouses, the systematic placement of direct reachable waypoints does not even induce non-optimality. Since waypoints are placed on crossings, globally optimal trajectories consist of a composition of locally optimal trajectories with direct sight.
- We restrict the movement to one dimension at a time. In our use case, this constraint is always met, since storage compartments and shelves are orthogonal.

The above mentioned constrains are only necessary when optimality has to be guaranteed. When suboptimal motion is acceptable, the assumptions can be relaxed. The lack of parameters is an advantage, compared to classic approaches like PID control where parameters need to be tuned.

II. Related Work

Trajectory generation for MAVs is an active field in scientific research. The state of the art can be subdivided into approaches that either assume it is possible to change the MAV velocity instantaneously (e.g., Nieuwenhuisen et al. [1]), or fully incorporate the dynamics of the MAV. As our approach counts to the latter, we focus on dynamic trajectory generation approaches. Trajectories are either generated offline in a more long-term application, or in a short-term way for online usage in real-time feedback systems like Model Predictive Control (MPC).

The first category of methods includes the work of Motonaka et al. [2], who use kinodynamic planning. Due to the complex model parameter estimation and the computationally expensive nature of kinodynamic planning, the practical applicability of this approach is limited. Richter et al. [3] use iterative refinement of polynomials after finding a valid trajectory via straight-line rapidly exploring random trees (RRT). As the RRT needs $3\,\mathrm{s}$ to compute ($t < 1\,\mathrm{ms}$ for the iterative polynomial refinement), also their approach is not real-time capable.

A hybrid system, using offline and online trajectory generation was developed by Brescianini et al. [4]. They use the ACADO toolbox to generate pole-throwing maneuvers offline and execute them open loop. Simultaneously, they generate a minimum snap real-time trajectory for catching the same pole. The catching maneuver does not consider state constraints, e.g., maximum velocity.

MPC-like trajectory generators have been developed by several groups. For example, Tomic et al. [5], Van Loock et al. [6], Ritz at al. [7], and Kahale et al. [8] use a similar first-principles model to our work. The trajectory generation problem is solved via numerical optimization by the first three works, and with a MATLAB nonlinear program solver (direct collocation approach) by the latter work. Each of these works relies on the differential flatness of the MAV model. Hehn et al. [9] replan time-optimal trajectories at every time step, making the method similar to an MPC. The approach assumes direct line of sight between current and final state. Furthermore, it is not able to incorporate velocity constraints, and is only able to generate trajectories targeting the vehicle at zero speed.

The works of Achtelik et al. [10] and Mellinger et al. [11] use N^{th} order polynomials with more degrees of freedom than equality constraints and a numerical solver to generate MPC-like trajectories. Boeuf et al. [12] employ splines in a decoupled global and local planner, resulting in a bang-singular-bang optimal trajectory, similar to our work. The solution, however, is not analytical.

In their works [13] and [14], Mueller et al. present a trajectory generator, capable of approaching the full target state (position, velocity, and acceleration). Analogue to our approach, they use jerk (respectively the rotational velocity ω) as system input, but the convex optimization problem is solved numerically.

In [15], Kröger presents a state-to-state solver for polynomial trajectories of robot manipulators with acceleration (Type II) or jerk (Type IV) as input. The algorithm runs with a typical cycle time of $t \leq 1\,\mathrm{ms}$, but no worst case or guaranteed cycle time is given. The Type IV solver, corresponding to the work presented here, is not publicly available.

To the knowledge of the authors, there exists no optimal analytical trajectory generation method for MAVs. Existing analytical methods, like PID-trajectory tracking, lead to non-optimality and cannot handle nonlinearities like state constraints. Our approach extends our previous work [16].

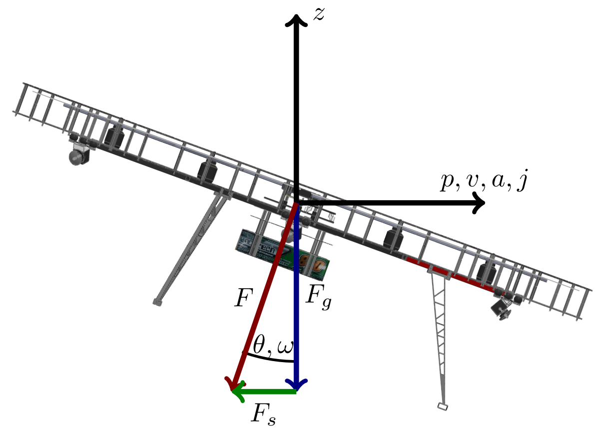

Fig. 2. Forces and state variables of the 2D hexacopter model used in this paper.

The key contributions of our paper are:

- derivation of a simplified first-principles model of the MAV,
- analytically generating smooth trajectories that involve not only position, but velocity, pitch, and rotational velocity,
- evaluation in a realistic scenario, in comparison to a state of the art cascaded PID position-velocity-attitude loop.

III. System Model

Our MAV design is a hexacopter with a frame surrounding the rotor plane. Fig. 1 shows it in a typical indoor environment—a warehouse. A detailed description of the used MAV including sensors, onboard processing systems and methods for state estimation and control can be found in [17].

A. Equations of Motion

We assume the MAV to follow rigid body dynamics and simplify it as a point mass with jerk j as system input. Following Newtons second law, the system is a triple integrator in each dimension with position p, velocity v, acceleration a, and jerk j. Since every point-to-point motion can be projected on a 2D plane by the unit vectors p and z, we use the 2D hexacopter model in Fig. 2. Furthermore, the MAV can be rotated by angle θ (pitch) with rotational velocity ω (pitch rate). The corresponding state vector $\boldsymbol{x}$ (Eq. 1) and system equations (Eq. 2) are

$$\boldsymbol{x} = \begin{pmatrix} p \\ v \\ a \end{pmatrix}, \tag{1}$$

$$\dot{p} = v, \qquad \dot{v} = a, \qquad \dot{a} = j. \tag{2}$$

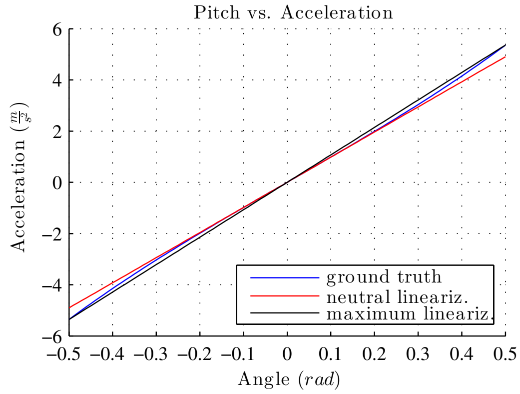

Fig. 3. Linearization of the horizontal acceleration. While retaining a constant vertical acceleration of $a = g = 9.81\,\frac{\mathrm{m}}{\mathrm{s}^2}$, the true horizontal acceleration (blue) is linearized either around neutral attitude (red) or maximum pitch (black).

We assume a constant vertical acceleration of $g = 9.81\,\frac{\mathrm{m}}{\mathrm{s}^2}$ and a constant height z. Therefore, the vertical thrust is constant $F_g = m \cdot g$. The assumption of a constant vertical acceleration simplifies the navigation problem to one dimension.

We further assume the rotational velocity $\dot{\theta} = \omega$ to be the direct control input to the system. Underlying dynamics like motor inertia are assumed to be neglectable and/or perfectly controlled. This assumption is justified since ω and all inherent underlying dynamics are controlled by a high-bandwidth, high gain controller, using gyroscope feedback. Since ω is tracked with a rate of $400\,\mathrm{Hz}$, the impact of underlying dynamics can be neglected.

Sec. III-B describes how the true control input ω is mapped to state input j. We want to emphasize that this model needs no parameters and that especially the estimation of hard-to-find parameters like motor constants or inertia tensor is not needed in our approach.

B. Model Linearization

Fig. 3 shows the horizontal acceleration a caused by force F_s in relation to deflection θ. It was obtained by

$$a = g \cdot tan(\theta). \tag{3}$$

The total thrust F of the MAV evolves with

$$F = \frac{F_g}{\cos(\theta)}, \tag{4}$$

to compensate for gravity.

Often, the small angle assumption linearizes around the hover point. Instead, here we use linearization around the maximum pitch, as time-optimal trajectories have the property to maximize the usage of extreme control inputs (Pontryagin's minimum principle). In Sec. IV-B, this behavior will be detailed. We evaluate our linearization method in Sec. V-A.

With a maximum pitch of $\theta_{max} = 0.5\,\mathrm{rad}$ and linearization

$$a = (\frac{g \cdot tan(\theta_{max})}{\theta_{max}}) \cdot \theta, \tag{5}$$

the state vector $\boldsymbol{x}$, and the system equations can be substituted by observable quantities:

$$\boldsymbol{x} = \begin{pmatrix} p \\ v \\ \theta \end{pmatrix}, \tag{6}$$

$$\dot{p} = v, \qquad \dot{v} = a = 10.72 \cdot \theta, \qquad \dot{\theta} = \omega. \tag{7}$$

IV. Trajectory Generation

We state the trajectory generation problem as follows:

Find a trajectory $\mathcal{T}(t)$ that satisfies state constraints (IV-A) and dynamic constraints (IV-C, IV-D) such as smoothness of the position p, velocity v, and pitch θ and minimizes time t.

The trajectory generation shall be real-time capable, and the generated trajectory time- and energy-optimal. We want to emphasize here that our simplified model is differentially flat.

A. State Constraints

The trajectory generation method has to make sure that the following state and input constraints are never violated:

$$v_{min} \leq v \leq v_{max}, \tag{8}$$
$$\theta_{min} \leq \theta \leq \theta_{max}, \tag{9}$$
$$\omega_{min} \leq \omega \leq \omega_{max}. \tag{10}$$

B. Trajectory Composition

In [18], Pontryagin shows that there exists a particular type of trajectories that satisfy time-optimality. The so called Pontryagin's minimum principle states that extreme control inputs minimize the time to drive a dynamic system from one state to another. In optimal control theory, a method that maximizes control input that lies on the edge of the control envelope is know as bang-bang or bang-singular-bang control.

Instead of specifying the Hamiltonian and minimizing it, we follow a more intuitive procedure. We decompose the trajectory into seven parts that either yield maximum pitch rate ($\omega = \omega_{max}$), zero pitch rate ($\omega = 0$), or minimum pitch rate ($\omega = \omega_{min}$). This is illustrated in Fig. 4 for the following example.

The optimal strategy for starting from state space point $\boldsymbol{x} = (0, 0, 0)$ (starting at the origin, zero velocity, zero pitch) with target $\boldsymbol{x}_{wayp} = (5, 0, 0)$ (position = $5\,\mathrm{m}$, zero velocity, zero pitch) in state space is:

I) Start to pitch forward with maximum pitch rate $\omega = \omega_{max}$ until you reach maximum pitch $\theta = \theta_{max}$,
II) stop pitching $\omega = 0$ and stay at maximum pitch $\theta = \theta_{max}$ until your velocity is 'fast enough',
III) start to pitch backward with minimum pitch rate $\omega = \omega_{min}$ until you reach zero pitch $\theta = 0$ and exact the maximum allowed velocity $v = v_{max}$,

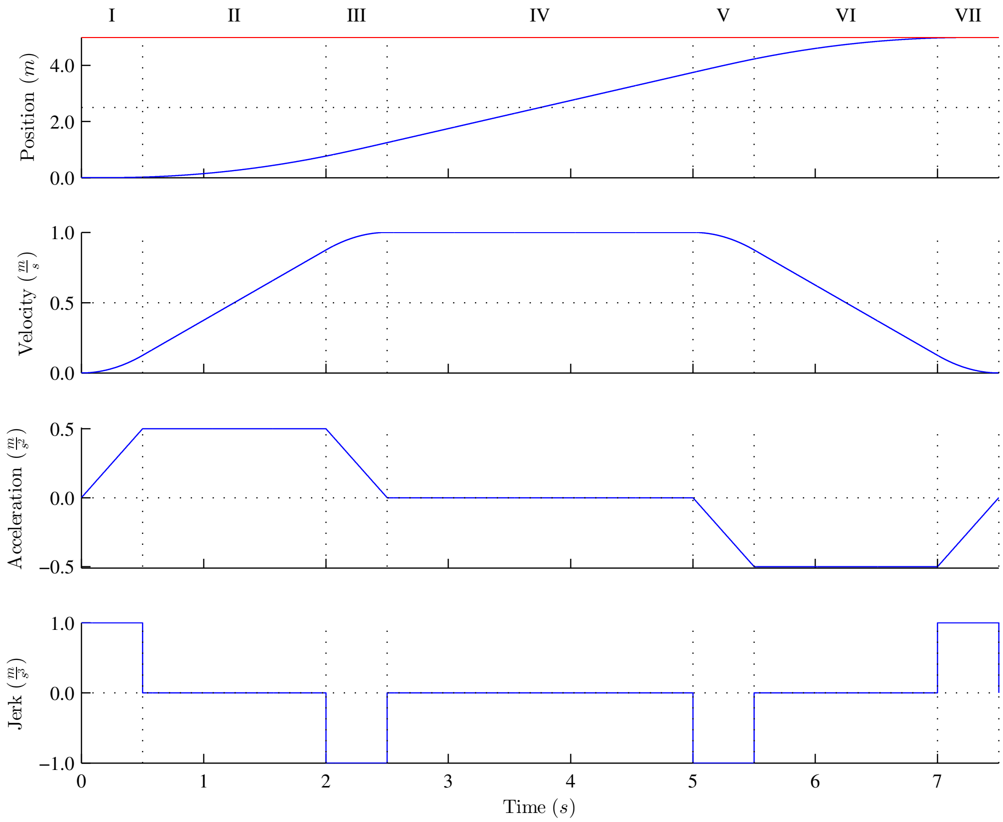

Fig. 4. This time-optimal trajectory was generated with our method. Starting from state $\boldsymbol{x} = (0, 0, 0)$ (starting at the origin, zero velocity, zero pitch), it brings the simulated MAV to state $\boldsymbol{x}_{wayp} = (5, 0, 0)$ (5 m, zero velocity, zero pitch). The trajectory complies with the following constraints: $v_{max} = -v_{min} = 1\,\frac{\mathrm{m}}{\mathrm{s}}$, $\theta_{max} = -\theta_{min} = 0.051\,\mathrm{rad} \Leftrightarrow a_{max} = -a_{min} = 0.5\,\frac{\mathrm{m}}{\mathrm{s}^2}$, $\omega_{max} = -\omega_{min} = 0.102\,\frac{\mathrm{rad}}{\mathrm{s}} \Leftrightarrow j_{max} = -j_{min} = 1\,\frac{\mathrm{m}}{\mathrm{s}^3}$. The calculated switching times are $t_1 = 0.5\,\mathrm{s}$, $t_2 = 1.5\,\mathrm{s}$, $t_3 = 0.5\,\mathrm{s}$, $t_4 = 2.5\,\mathrm{s}$, $t_5 = 0.5\,\mathrm{s}$, $t_6 = 1.5\,\mathrm{s}$, and $t_7 = 0.5\,\mathrm{s}$.

IV) stop pitching $\omega = 0$ and stay at zero pitch $\theta = 0$ until traveled distance is 'far enough',

V) start to pitch backward with minimum pitch rate $\omega = \omega_{min}$ until you reach minimum pitch $\theta = \theta_{min}$,

VI) stop pitching $\omega = 0$ and stay at minimum pitch $\theta = \theta_{min}$ until your velocity is 'slow enough', and

VII) start to pitch forward with maximum pitch rate $\omega = \omega_{max}$ until you reach your desired pitch $\theta = \theta_{wayp}$. At this point you should exactly reach the desired state $\boldsymbol{x} = \boldsymbol{x}_{wayp}$.

C. First-order Conditions

For every part n of the composed trajectory, we formulate a system of three differential equations, reflecting the state at the end of the part, including position p, velocity v, and acceleration a:

$$a_n = a_{n-1} + \int_0^{t_n} j_n \,\mathrm{d}t, \tag{11}$$

$$v_n = v_{n-1} + \int_0^{t_n} a_n \,\mathrm{d}t, \tag{12}$$

$$p_n = p_{n-1} + \int_0^{t_n} v_n \,\mathrm{d}t. \tag{13}$$

Under the assumption of extreme control inputs, stated in Sec. IV-B, we can derive a system of non-differential equations:

$$a_1 = a_0 + t_1 \cdot j_{max}, \tag{14}$$

$$a_2 = a_1 + t_2 \cdot 0, \tag{15}$$

$$a_3 = a_2 + t_3 \cdot j_{min}, \tag{16}$$

$$a_4 = a_3 + t_4 \cdot 0, \tag{17}$$

$$a_5 = a_4 + t_5 \cdot j_{min}, \tag{18}$$

$$a_6 = a_5 + t_6 \cdot 0, \tag{19}$$

$$a_7 = a_6 + t_7 \cdot j_{max}, \tag{20}$$

$$v_1 = v_0 + t_1 \cdot a_0 + \frac{1}{2} \cdot t_1^2 \cdot j_{max}, \tag{21}$$

$$v_2 = v_1 + t_2 \cdot a_1 + \frac{1}{2} \cdot t_2^2 \cdot 0, \tag{22}$$

$$v_3 = v_2 + t_3 \cdot a_2 + \frac{1}{2} \cdot t_3^2 \cdot j_{min}, \tag{23}$$

$$v_4 = v_3 + t_4 \cdot a_3 + \frac{1}{2} \cdot t_4^2 \cdot 0, \tag{24}$$

$$v_5 = v_4 + t_5 \cdot a_4 + \frac{1}{2} \cdot t_5^2 \cdot j_{min}, \tag{25}$$

$$v_6 = v_5 + t_6 \cdot a_5 + \frac{1}{2} \cdot t_6^2 \cdot 0, \tag{26}$$

$$v_7 = v_6 + t_7 \cdot a_6 + \frac{1}{2} \cdot t_7^2 \cdot j_{max}, \tag{27}$$

$$p_1 = p_0 + t_1 \cdot v_0 + \frac{1}{2} \cdot t_1^2 \cdot a_0 + \frac{1}{6} \cdot t_1^3 \cdot j_{max}, \tag{28}$$

$$p_2 = p_1 + t_2 \cdot v_1 + \frac{1}{2} \cdot t_2^2 \cdot a_1 + \frac{1}{6} \cdot t_2^3 \cdot 0, \tag{29}$$

$$p_3 = p_2 + t_3 \cdot v_2 + \frac{1}{2} \cdot t_3^2 \cdot a_2 + \frac{1}{6} \cdot t_3^3 \cdot j_{min}, \tag{30}$$

$$p_4 = p_3 + t_4 \cdot v_3 + \frac{1}{2} \cdot t_4^2 \cdot a_3 + \frac{1}{6} \cdot t_4^3 \cdot 0, \tag{31}$$

$$p_5 = p_4 + t_5 \cdot v_4 + \frac{1}{2} \cdot t_5^2 \cdot a_4 + \frac{1}{6} \cdot t_5^3 \cdot j_{min}, \tag{32}$$

$$p_6 = p_5 + t_6 \cdot v_5 + \frac{1}{2} \cdot t_6^2 \cdot a_5 + \frac{1}{6} \cdot t_6^3 \cdot 0, \tag{33}$$

$$p_7 = p_6 + t_7 \cdot v_6 + \frac{1}{2} \cdot t_7^2 \cdot a_6 + \frac{1}{6} \cdot t_7^3 \cdot j_{max}. \tag{34}$$

D. Second-order Conditions

Since only 21 equations and 31 unknown variables are defined ($t_1, ..., t_7$, $a_0, ..., a_7$, $v_0, ..., v_7$, $p_0, ..., p_7$), we need second-order conditions to fully describe the trajectory. We therefore assume $\boldsymbol{x}$ to be the current state and $\boldsymbol{x}_{wayp}$ the target state. We also assume Eq. 41 and the constraints in Tab. I stated under 'Case 1':

$$a_0 = a, \tag{35}$$

$$v_0 = v, \tag{36}$$

$$p_0 = p, \tag{37}$$

$$a_7 = a_{wayp}, \tag{38}$$

$$v_7 = v_{wayp}, \tag{39}$$

$$p_7 = p_{wayp}, \tag{40}$$

$$a_3 = 0. \tag{41}$$

E. Feasibility

Since we make the assumption that the MAV reaches the maximum allowed velocity and pitch during the trajectory, this approach is not feasible for arbitrary configurations of $\boldsymbol{x}$ and $\boldsymbol{x}_{wayp}$. In contrast to other approaches that simply lower the maximum pitch rate ω_{max} to converge to a feasible trajectory, we generate eight different trajectory templates that follow a different policy each. An optimal trajectory could, e.g., be to only pitch forward and immediately pitch backwards—without the part of constant velocity in the middle. This leads to a substitution of second-order condition $v_3 = v_{max} \rightarrow t_4 = 0$ (Case 2). All permuted substitutions, necessary for all possible trajectory templates are shown in Tab. I.

We further mirror the trajectory templates to expand the reachable set to all possible state space configurations. So, with $j_1 = j_7 = j_{min}$, $j_3 = j_5 = j_{max}$ and $j_2 = j_4 = j_6 = 0$, we get 16 trajectory templates in total.

F. Analytical Solution

We use the MathWorks® MATLAB Symbolic Math Toolbox to find an analytical solution to the problem stated. The analytical solution for $t_1, ..., t_7$ for Case 1 is shown in the appendix (Eq. 44 – Eq. 50). For better readability, we refrain from substituting a and j, as this induces many constants into the equation. We show a numerical solution for $\boldsymbol{x} = (0, 0, 0)$ and $\boldsymbol{x}_{wayp} = (5, 0, 0)$ in Fig. 4. The corresponding pitch angles and the evolution of the state space are shown in Fig. 5 and Fig. 6, respectively.

We test for feasibility by simply evaluating all trajectory templates and choosing the one that does not violate any state, and causality constraints ($t_1, ..., t_7 \geq 0$).

Since only linear, quadratic, cubic, and quartic equations can be solved analytically (Abel-Ruffini theorem), our approach is limited to jerk as system input. Higher derivative input like snap prohibits the analytical solution. On first glance, this poses a limit to the proposed approach. In reality, since underlying dynamics are controlled by a high-bandwidth, high gain controller, it poses no limitation for the problem stated. E.g. Mellinger et al. [19] give insight into the dynamics of pitch control. We further justify this design choice in Sec. V-E.

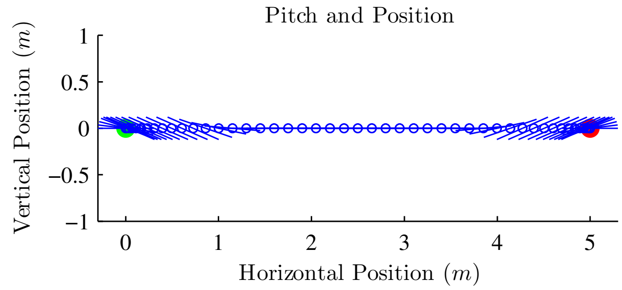

Fig. 5. Trajectory when flying from $\boldsymbol{x} = (0, 0, 0)$ (green marker) to $\boldsymbol{x}_{wayp} = (5, 0, 0)$ (red marker). The pitch angle is depicted by the line. Parameters for this simulation where chosen to be $v_{max} = -v_{min} = 1\,\frac{\text{m}}{\text{s}}$, $\theta_{max} = -\theta_{min} = 0.051\,\text{rad} \Leftrightarrow a_{max} = -a_{min} = 0.5\,\frac{\text{m}}{\text{s}^2}$, $\omega_{max} = -\omega_{min} = 0.102\,\frac{\text{rad}}{\text{s}} \Leftrightarrow j_{max} = -j_{min} = 1\,\frac{\text{m}}{\text{s}^3}$, blue markers are placed every 150 ms. For better readability, the pitch angle is magnified 8 times.

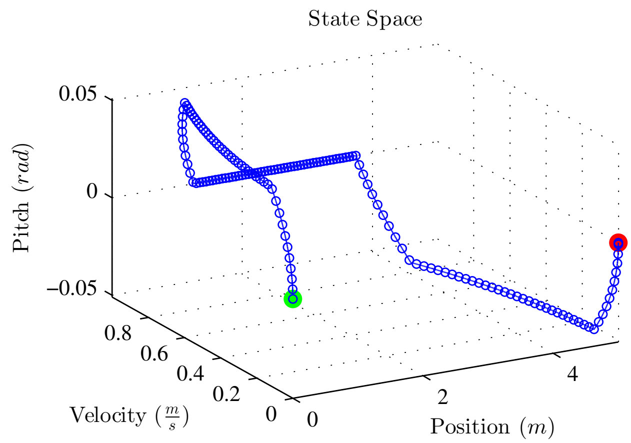

Fig. 6. Evolution of the state space when flying from $\boldsymbol{x} = (0, 0, 0)$ (green marker) to $\boldsymbol{x}_{wayp} = (5, 0, 0)$ (red marker). Parameters for this simulation where chosen to be $v_{max} = -v_{min} = 1\,\frac{\text{m}}{\text{s}}$, $\theta_{max} = -\theta_{min} = 0.051\,\text{rad} \Leftrightarrow a_{max} = -a_{min} = 0.5\,\frac{\text{m}}{\text{s}^2}$, $\omega_{max} = -\omega_{min} = 0.102\,\frac{\text{rad}}{\text{s}} \Leftrightarrow j_{max} = -j_{min} = 1\,\frac{\text{m}}{\text{s}^3}$, blue markers are placed every 50 ms.

TABLE I

SECOND-ORDER CONDITIONS

Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
$\theta_1 = \theta_{max}$	$\theta_1 = \theta_{max}$	$t_2 = 0$	$\theta_1 = \theta_{max}$	$t_2 = 0$	$t_2 = 0$	$\theta_1 = \theta_{max}$	$t_2 = 0$
$\theta_5 = \theta_{min}$	$\theta_5 = \theta_{min}$	$\theta_5 = \theta_{min}$	$t_6 = 0$	$t_6 = 0$	$\theta_5 = \theta_{min}$	$t_6 = 0$	$t_6 = 0$
$v_3 = v_{max}$	$t_4 = 0$	$v_3 = v_{max}$	$v_3 = v_{max}$	$v_3 = v_{max}$	$t_4 = 0$	$t_4 = 0$	$t_4 = 0$

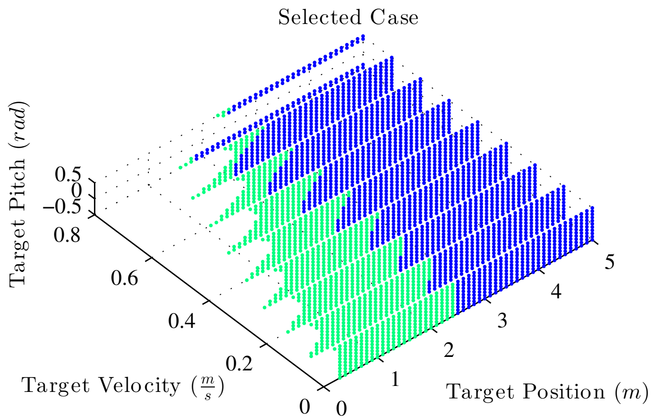

Fig. 7. Starting from $\boldsymbol{x} = (0, 0, 0)$, it can be seen where Case 1 (blue) or Case 2 (aquamarine) is valid for a given configuration of $\boldsymbol{x}_{wayp}$.

The equations present in our approach can be solved by e.g., PQ-formula, Cardano's method, Ferrari's method, or by an automatic solver like the one used herein.

Fig. 7 shows the cases used, dependent on the target state $\boldsymbol{x}_{wayp}$ when starting from $\boldsymbol{x} = (0, 0, 0)$.

V. RESULTS

A. Linearization

In Fig. 3, it can be seen that the worst-case error (when linearizing with a realistic maximum pitch of $\theta_{max} = 0.5\,\text{rad}$) is well below 10% of the true value, a quality usually not achieved with model parameter identification.

Since the actual error heavily depends on the trajectory, it is impossible to generalize the error metric to all trajectories. Exemplarily, we evaluate the error of the trajectory shown in Fig. 4.

As seen in Fig. 8, the maximum error magnitude of $170\,\frac{\mu\text{m}}{\text{s}^2}$ is three orders of magnitude smaller then the maximum nominal acceleration of $0.5\,\frac{\text{m}}{\text{s}^2}$. When the MAV is in the state of either zero acceleration (hover condition) or maximum acceleration, the error is completely eliminated when linearizing with our method. This yields to a better linearization-induced average absolute error per trajectory

$$e_a = \frac{\int_0^{t_1+\ldots+t_7} |a_{gt} - a_{lin}|\,\mathrm{d}t}{t_1 + \ldots + t_7}, \tag{42}$$

shown in Tab. II.

It can be seen that trajectories where times with zero or maximum acceleration exceed transient times ($t_2+t_4+t_6 \gg t_1 + t_3 + t_5 + t_7$) show smaller linearization errors. This is

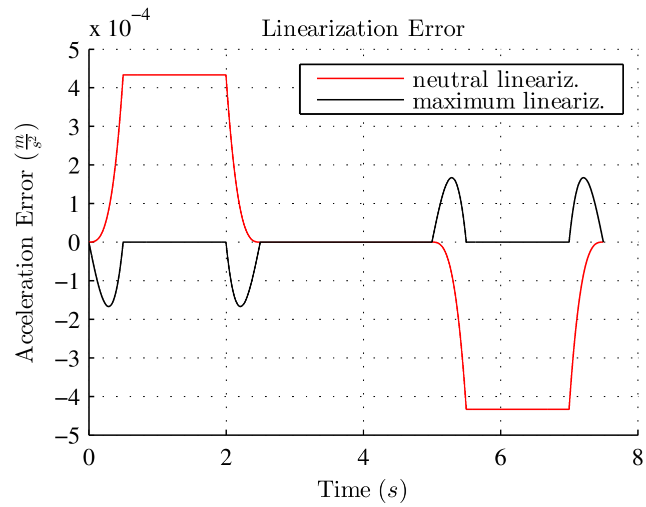

Fig. 8. We compare the progress of the acceleration error with linearization around the hover point and our linearization method. Here we simulate the trajectory shown in Fig. 4

valid for most trajectories as realistic pitch rates are usually in the range of $\omega \gg 5\,\frac{\text{rad}}{\text{s}}$.

In general, the relative linearization error is small in comparison to nominal accelerations and thus can be neglected ($0.5\,\text{mm}$ error at a $5\,\text{m}$ trajectory).

B. Energy Optimality

Under all trajectories that drive the MAV from the start- to the target state and satisfy state- and dynamic constraints, there exist an infinite subset that minimizes the times, the MAV accelerates and decelerates. This means the MAV is not decelerating and then accelerating again without reason. A reason would be to prevent violation of a constraint (e.g., not to overshoot over the target). In this subset there is a one specific trajectory that lies on the edge of the control envelope. This specific trajectory is the one our method generates.

We define the energy used by a trajectory as the integral of the total thrust F

$$E = \int_0^{t_1+\ldots+t_7} F\,\mathrm{d}t, \tag{43}$$

over the whole trajectory.

Fig. 9 shows the energy used for a set of trajectories with different maximum velocities and accelerations. It can be seen that the function is convex and that trajectories with larger accelerations and velocities consume less energy.

TABLE II

AVERAGE LINEARIZATION INDUCED ERROR

Linearization Technique	Average Absolute Error in $\frac{\mathrm{mm}}{\mathrm{s}^2}$	Final Position in m	Rel. Position Error in ppm
Neutral linearization	202.08	5.00381	762.69
Maximum linearization	**28.91**	4.99945	**-108.31**

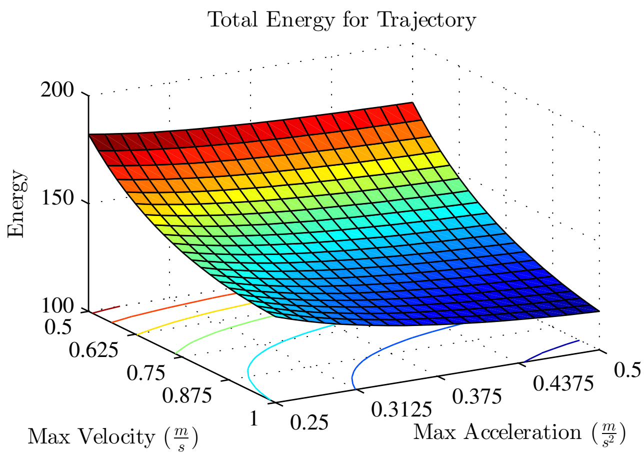

Fig. 9. Here we simulate the trajectory shown in Fig. 4 with different maximum accelerations and velocities. It can be seen that the energy function is convex and that it is always more energy efficient to accelerate faster and to fly faster. The trajectories generated by our method lie on the edge of the control envelope, and thus are energy-optimal. We assume the MAV to weight $m = 1.5\,\mathrm{kg}$.

The energy that is used to compensate for gravity and thus keep the MAV airborne grows linear with the total time of a trajectory. Consequently, slow trajectories need more energy. Thus, although faster trajectories consume more energy for acceleration and deceleration, they consume less energy overall, due to a shorter execution time.

Since our method always generates time optimal trajectories, these trajectories are also energy-optimal in terms of the total thrust intergral.

C. *Classic PID Control*

In Sec. IV-B we show that for time-optimal control, it is necessary for the control input to lie on the edge of the control envelope. Since classic PID control does so only in a special case (with saturated control input $k \to \infty$), it does not follow this so called Pontryagin's minimum principle. When saturating the control input however, classic PID control will overshoot, since the target acceleration will become negative (braking the MAV) only when the MAV has already reached the target position.

In contrast to our method, low-level constraints are not propagated into higher levels of the cascade. Thus when saturating, e.g., the acceleration setpoint, the next higher level (velocity loop) suddenly has limited control authority. This can either lead to overshoot, or slow transient responses, depending on the parameterization. While our approach exploits model knowledge to predict the evolution of the MAV state and thus can incorporate nonlinearities, classic control theory is limited to linear control. Thus, under the assumption of nonlinearities, classic PID control can under no conditions be faster than our time-optimal approach.

Classic cascaded PID control also relies on the proper parameterization of at least three parameters, while our method works parameterless.

Another drawback of the classic approach is the unability to specify the full target state of the MAV. Here, the position setpoint determines the velocity setpoint which then determines the acceleration (pitch) setpoint. The strictly hierarchical property and missing model knowledge makes it impossible to specify side conditions like target velocities and accelerations.

D. *Computation Time*

In order to prove the real-time capability of our approach, we measured the time to derive the switching times $t_1, ..., t_7$ by the equations stated at Case 1. We found that the whole process takes $t_{comp} \ll 1\,\mathrm{\mu s}$ on a standard laptop computer. This shows that the method is suitable to be running even on small flight control computers with appropriate rates.

Since analytical solutions to high order polynomials tend to become very complex, we analyzed the complexity of the results. Tab. III shows the approximate number of mathematical operations that is necessary to solve the problem. It can be seen that Case 6 – Case 8 are significantly more complex then Case 1 – Case 5.

Computation time evolves approximately linear with the number of mathematical operation. Case 6 is approximately 6504 times as complex as Case 1, so even the most complex Case 6 should be solvable in milliseconds. With realistic pitch rates, Case 6, 7 and 8 only occur very close to the target waypoint, when the there is no time of constant velocity ($t_4 = 0$) and either no time of constant acceleration (Case 6, $t_2 = 0$), constant deceleration (Case 7, $t_6 = 0$), or both (Case 8, $t_2 = 0$, $t_6 = 0$).

Case 8 only occurs in the most extreme conditions of the trajectory, very close to the target. It can be seen that when starting from $\boldsymbol{x} = (0, 0, 0)$, Case 8 is only valid when the target position is in a radius with $r \leq 3.12\,\mathrm{cm}$ (assuming realistic parameters of $v_{max} = -v_{min} = 1.5\,\frac{\mathrm{m}}{\mathrm{s}}$, $\theta_{max} = -\theta_{min} = 0.25\,\mathrm{rad}$, $\omega_{max} = -\omega_{min} = 1\,\frac{\mathrm{rad}}{\mathrm{s}}$ with $v_{wayp} = 0\,\frac{\mathrm{m}}{\mathrm{s}}$, and $a_{wayp} = 0\,\frac{\mathrm{m}}{\mathrm{s}^2}$).

Since Cases 6, 7 and 8 nearly never occur in realistic data, we refrain from solving them explicitly.

E. *EuRoC Simulator*

We use the RotorS MAV simulator [20] which was developed for Challenge 3 of the European Robotics Challenges

TABLE III

COMPLEXITY ANALYSIS

Time	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
t_1	2	3	58	3	58	2	3	434540
t_2	20	242	0	14	0	0	22586	0
t_3	1	1	47	1	47	1433787	1	217269
t_4	182	0	102	433	176	0	0	0
t_5	1	1	1	53	54	1	11283	217269
t_6	21	242	13	0	0	62319	0	0
t_7	3	2	2	64	65	2	11281	144812

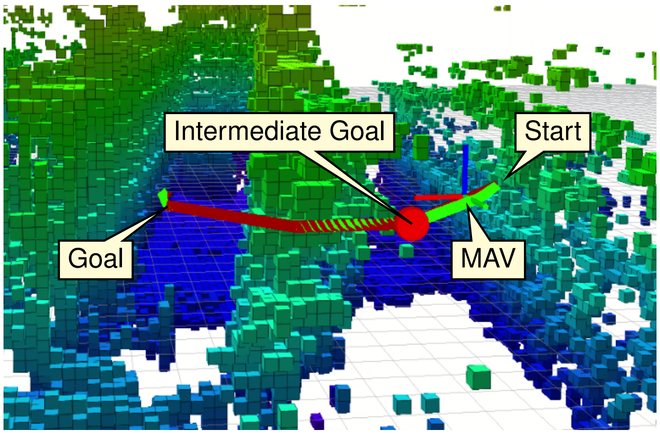

Fig. 10. Our method is evaluated within the RotorS MAV simulator. We plan obstacle-free paths (green) from the MAVs starting location to the goal in an allocentric gridmap of a warehouse (see Fig. 1). After pruning (red), we employ our method to traverse the path. Here, the MAV (depicted by the axes) already executed half of the trajectory to the intermediate waypoint. Color encodes height.

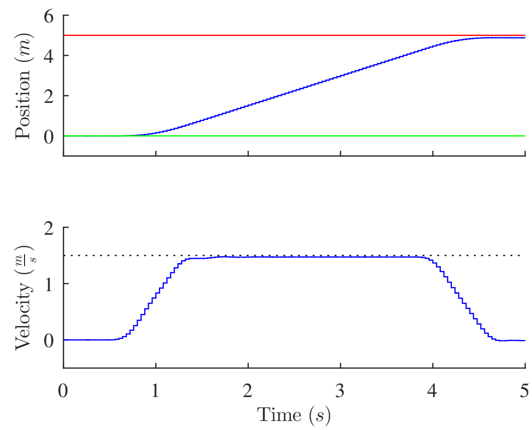

Fig. 11. Nonlinear simulated flight from state $\boldsymbol{x} = (0,0,0)$ to state $\boldsymbol{x}_{wayp} = (5,0,0)$ by RotorS. Parameters for the trajectory are chosen to be $v_{max} = 1.5\,\frac{\mathrm{m}}{\mathrm{s}}$, $\theta_{max} = -\theta_{min} = 0.25\,\mathrm{rad}$, $\omega_{max} = -\omega_{min} = 1\,\frac{\mathrm{rad}}{\mathrm{s}}$.

(EuRoC) to prove the applicability of our method to nonlinear models with realistic parameters. A typical test scenario is shown in Fig. 10.

Since our method only allows direct waypoint flight, we employ a hierarchy of planners. First, we plan obstacle-free paths in an allocentric 3D gridmap of the environment and 26-neighborhood with A*. Details can be found in [21]. After that, we employ the Ramer-Douglas-Peucker algorithm to prune the planned path. We then use our method to plan trajectories connecting the intermediate waypoints of the path.

Due to the special characteristics of our environment (orthogonal walls, mostly vertical structures), we can guarantee optimality although the target waypoint is not in direct line of sight. Since waypoints are placed on crossings, globally optimal trajectories consist of a composition of locally optimal trajectories with direct sight. Thus, we can even relax the assumption made in Sec. I.

By adjusting the waypoint radius, MAV flight can be either precise or fast. When the waypoint radius is small, the MAV stops at intermediate waypoints and reaches them exactly. If the waypoint radius is large, the MAV passes the waypoint at constant maximum velocity, but does not reach it exactly. In this way, a balance between precision and speed at intermediate waypoints can be achieved.

Fig. 11 shows the corresponding position and velocity graphs for a flight from state $\boldsymbol{x} = (0,0,0)$ to state $\boldsymbol{x}_{wayp} = (5,0,0)$. Instead of replanning at every control cycle, we only plan the trajectory once. By doing so, we can identify errors induced by the model linearization and the underlying dynamics. For this experiment, rotor drag is switched off. It can be seen that possibly because of linearization induced errors, the maximum velocity is not exactly reached ($v_{max} = 1.4767\,\frac{\mathrm{m}}{\mathrm{s}} \neq 1.5\,\frac{\mathrm{m}}{\mathrm{s}}$). This results in a positional error at the end of the trajectory of $p_{wayp} = 4.8807\,\mathrm{m} \neq 5\,\mathrm{m}$. We want to emphasize here that the trajectory was executed open loop so that the error of 2.4% could be easily reduced by replanning the trajectory during execution.

Videos of this simulation and other experiments can be found on our website[1].

VI. CONCLUSIONS

The paper proposes an analytical time- and energy-optimal trajectory generation method for MAVs that is able to run in real-time and thus can be used as MPC.

Starting from a simple parameterless first-principles model of the MAV, we compute optimal switching times for the rotational velocity of the MAV, respecting state, and input constraints.

With the ability to specify the full state of the MAV, it is possible to target moving or even accelerating waypoints like, e.g., a moving landing platform.

[1] `http://www.ais.uni-bonn.de/videos/ICUAS_2016`

We evaluate the effectiveness of the proposed approach, comparing it to the state of the art cascaded control loop for position, velocity, and attitude. The classic design can violate high-level constraints, as low-level state constraints are not propagated into higher levels of the cascade.

At the moment, optimality is only given for exclusively horizontal movement and waypoints that have either direct line of sight or lie within an orthogonal environment. Since our use case and many other use cases deal with flying in man-made environments that mostly consist of vertical structures, these constraints are often met. Nevertheless, in future work we want to relax the assumption of constant height and extend the method to full 3D movements.

We show that in general, our approach is capable of dealing with arbitrary input and state constraints. In future work we want to evaluate if it is possible to model asymmetric model properties, e.g., acceleration is faster then deceleration because of drag.

At the moment, we test for feasability by evaluating all trajectory templates and choosing the one that is valid. This poses no computational problem, since the method is very fast, but it would be more elegant to model the problem with, e.g., a decision tree with state- and dyanamic thresholds.

The method can be employed as is, or build upon.

REFERENCES

[1] M. Nieuwenhuisen and S. Behnke, "Layered mission and path planning for MAV navigation with partial environment knowledge," in *Proc. of the Int. Conf. on Intelligent Autonomous Systems (IAS)*, 2014.

[2] K. Motonaka, K. Watanabe, and S. Maeyama, "Kinodynamic motion planning and control for an x4-flyer using anisotropic damping forces," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[3] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for agressive quadrotor flight in dense indoor environments," in *Proc. of the Int. Symposium of Robotics Research (ISRR)*, 2013.

[4] D. Brescianini, M. Hehn, and R. D'Andrea, "Quadrocopter pole acrobatics," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[5] T. Tomić, M. Maier, and S. Haddadin, "A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasability verification," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[6] W. V. Loock, G. Pipeleers, and J. Swevers, "Time-optimal quadrotor flight," in *Proc. of the European Control Conference (ECC)*, 2013.

[7] R. Ritz, M. Hehn, S. Lupashin, and R. D'Andrea, "Quadrocopter performance benchmarking using optimal control," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[8] E. Kahale, P. C. Garcia, and Y. Bestaoui, "Minimum time reference trajectory generation for an autonomous quadrotor," in *Proc. of the Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2014.

[9] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," in *IFAC World Congress*, 2011.

[10] M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Inversion based direct position control and trajectory following for micro aerial vehicles," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[11] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[12] A. Boeuf, J. Cortés, R. Alami, and T. Siméon, "Planning agile motions for quadrotors in constrained environments," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[13] M. W. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," in *Proc. of the European Control Conference (ECC)*, 2011.

[14] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasability verification," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[15] T. Kröger, "Opening the door to new sensor-based robot applications - the reflexxes motion libraries," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[16] M. Beul, S. Behnke, and R. Worst, "Nonlinear model-based 2D-position control for quadrotor UAVs," in *Proceedings of the Joint Int. Symposium on Robotics (ISR) and the German Conference on Robotics (ROBOTIK)*, 2014.

[17] M. Beul, N. Krombach, Y. Zhong, D. Droeschel, M. Nieuwenhuisen, and S. Behnke, "A high-performance MAV for autonomous navigation in complex 3D environments," in *Proc. of the Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2015.

[18] L. Pontryagin, *Mathematical Theory of Optimal Processes*, ser. Classics of Soviet Mathematics. Taylor & Francis, 1987.

[19] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, p. 0278364911434236, 2012.

[20] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Cham: Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26054-9_23

[21] M. Nieuwenhuisen, D. Droeschel, M. Beul, and S. Behnke, "Autonomous navigation for micro aerial vehicles in complex GNSS-denied environments," *Journal of Intelligent & Robotic Systems*, pp. 1–18, 2015.

APPENDIX
ANALYTICAL SOLUTION FOR CASE 1

$$t_1 = \frac{a_{max} - a}{j_{max}} \tag{44}$$

$$t_2 = \frac{a_{max}^2 \cdot j_{max} - a_{max}^2 \cdot j_{min} + a^2 \cdot j_{min} + 2 \cdot j_{min} \cdot j_{max} \cdot v_{max} - 2 \cdot j_{min} \cdot j_{max} \cdot v}{2 \cdot a_{max} \cdot j_{min} \cdot j_{max}} \tag{45}$$

$$t_3 = -\frac{a_{max}}{j_{min}} \tag{46}$$

$$t_4 = -\frac{A + B + C + D + E + F + G + H + I}{J} \tag{47}$$

$$A = a_{min} \cdot a_{max}^4 \cdot j_{min}^2 - a_{min}^4 \cdot a_{max} \cdot j_{min}^2 - a_{min} \cdot a_{max}^4 \cdot j_{max}^2 + a_{min}^4 \cdot a_{max} \cdot j_{max}^2 - 3 \cdot a_{min} \cdot a^4 \cdot j_{min}^2$$

$$B = 3 \cdot a_{max} \cdot a_{wayp}^4 \cdot j_{min}^2 + 8 \cdot a_{min} \cdot a_{max} \cdot a^3 \cdot j_{min}^2 - 8 \cdot a_{min} \cdot a_{max} \cdot a_{wayp}^3 \cdot j_{min}^2$$

$$C = -6 \cdot a_{min} \cdot a_{max}^2 \cdot a^2 \cdot j_{min}^2 + 6 \cdot a_{min}^2 \cdot a_{max} \cdot a_{wayp}^2 \cdot j_{min}^2 + 12 \cdot a_{min} \cdot j_{min}^2 \cdot j_{max}^2 \cdot v_{max}^2$$

$$D = -12 \cdot a_{max} \cdot j_{min}^2 \cdot j_{max}^2 \cdot v_{max}^2 - 12 \cdot a_{min} \cdot j_{min}^2 \cdot j_{max}^2 \cdot v^2 + 12 \cdot a_{max} \cdot j_{min}^2 \cdot j_{max}^2 \cdot v_{wayp}^2$$

$$E = 24 \cdot a_{min} \cdot a_{max} \cdot j_{min}^2 \cdot j_{max}^2 \cdot p - 24 \cdot a_{min} \cdot a_{max} \cdot j_{min}^2 \cdot j_{max}^2 \cdot p_{wayp}$$

$$F = -12 \cdot a_{min} \cdot a_{max}^2 \cdot j_{min} \cdot j_{max}^2 \cdot v_{max} + 12 \cdot a_{min}^2 \cdot a_{max} \cdot j_{min} \cdot j_{max}^2 \cdot v_{max}$$

$$G = 12 \cdot a_{min} \cdot a_{max}^2 \cdot j_{min}^2 \cdot j_{max} \cdot v - 12 \cdot a_{min}^2 \cdot a_{max} \cdot j_{min}^2 \cdot j_{max} \cdot v_{wayp}$$

$$H = 12 \cdot a_{min} \cdot a^2 \cdot j_{min}^2 \cdot j_{max} \cdot v - 12 \cdot a_{max} \cdot a_{wayp}^2 \cdot j_{min}^2 \cdot j_{max} \cdot v_{wayp}$$

$$I = -24 \cdot a_{min} \cdot a_{max} \cdot a \cdot j_{min}^2 \cdot j_{max} \cdot v + 24 \cdot a_{min} \cdot a_{max} \cdot a_{wayp} \cdot j_{min}^2 \cdot j_{max} \cdot v_{wayp}$$

$$J = 24 \cdot a_{min} \cdot a_{max} \cdot j_{min}^2 \cdot j_{max}^2 \cdot v_{max}$$

$$t_5 = \frac{a_{min}}{j_{min}} \tag{48}$$

$$t_6 = -\frac{a_{min}^2 \cdot j_{max} - a_{min}^2 \cdot j_{min} + a_{wayp}^2 \cdot j_{min} + 2 \cdot j_{min} \cdot j_{max} \cdot v_{max} - 2 \cdot j_{min} \cdot j_{max} \cdot v_{wayp}}{2 \cdot a_{min} \cdot j_{min} \cdot j_{max}} \tag{49}$$

$$t_7 = -\frac{a_{min} - a_{wayp}}{j_{max}} \tag{50}$$