Quadrocopter Trajectory Generation and Control

Markus Hehn [*] **Raffaello D'Andrea** [*]

[*] *Institute for Dynamic Systems and Control, ETH Zurich*
(e-mail: {hehnm, rdandrea}@ethz.ch).

Abstract: An algorithm is presented that allows the calculation of flight trajectories for quadrocopters. Trajectory feasibility constraints regarding the vehicle dynamics and input constraints are derived. They are then used in the planning algorithm to guarantee the feasibility of generated trajectories. The translational degrees of freedom of the quadrotor are decoupled, and time-optimal trajectories are found for each degree of freedom separately. The trajectory generation is fast enough to be performed online. Control inputs are calculated from the generated trajectory, and used to achieve closed-loop control similar to model predictive control. The trajectory generation and tracking performance is demonstrated in the ETH Zurich Flying Machine Arena testbed. Experimental results show good performance, with unmodeled aerodynamic effects causing trajectory deviations when decelerating from high speeds. Development potential for the future is highlighted, focusing on improving the performance and correcting for aerodynamic effects.

Keywords: Trajectory generation; Flying robots; Autonomous vehicle navigation, guidance and control; Mission planning and decision making; Motion control

1. INTRODUCTION

Due to their exceptional agility, quadrotor vehicles have emerged as popular miniature aerial vehicle platforms. Thanks to the off-center mounting of the propellers, extraordinarily fast rotational dynamics can be achieved. Typically high thrust-to-weight ratios allow large translational accelerations when not carrying a payload.

The controller design for near-hover operation of quadrotor vehicles has been covered in depth in many contributions (see, for example, How et al. [2008], and references therein). Most discussions assume that a trajectory exists and is to be followed. The question of the feasibility of a given trajectory has been studied more recently. Hoffmann et al. [2008], Cowling et al. [2007], and Bouktir et al. [2008] present algorithms that split the planning problem into two parts: First, trajectories containing no time information are calculated from a class of motion primitives (lines, polynomials, or splines). The trajectory is then parametrized in time by choosing the trajectory speed such that dynamic feasibility constraints are enforced. Lai et al. [2006] present a methodology using nonlinear programming and genetic algorithms to minimize the transition time between two given states.

In this paper, we develop a trajectory generation methodology that allows planning from any initial state to a target position. The main objectives of the approach presented in this paper are:

- The trajectories must be feasible under the dynamic and input constraints of the quadrotor vehicle,
- The generated trajectories should bring the vehicle to the target position as quickly as possible,
- The trajectory calculation must be fast enough to be used online at update rates around 50Hz, and
- It must be possible to generate an implicit feedback control law by replanning the trajectory at each controller update, and applying the control inputs of the first section of it.

This approach differs from the previous work presented above in that we seek to develop an algorithm that both directly incorporates the dynamic constraints of the vehicle at the planning stage, and is fast enough to allow real-time planning. This is achieved by conservatively approximating the feasibility constraints, such that the planning can be decoupled into three subproblems that can be solved efficiently. In particular, the approach is the following: The dynamic quadrocopter model has four control inputs, which are subject to saturation. We use the jerk of the three translational degrees of freedom as planning inputs. Feasibility of planned trajectories is

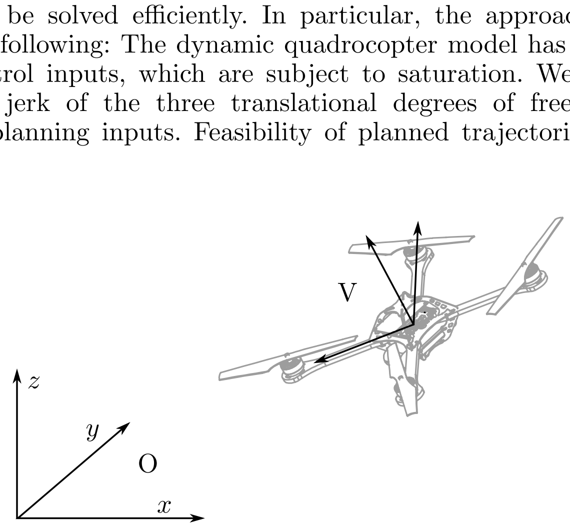

Fig. 1. The inertial coordinate system O and the vehicle coordinate system V.

guaranteed by limiting trajectory jerk and acceleration such that the actual control inputs do not saturate.

In Section 2, we introduce the dynamic quadrocopter model used in the trajectory generation. Feasibility constraints on trajectories are derived in Section 3. In Section 4, the planning problem is presented, simplified, and solved. Section 5 presents the implicit feedback control law resulting from running the trajectory generation algorithm at each controller update. The experimental setup and results are shown in Section 6. Conclusions are drawn in Section 7, along with an outlook for future research.

2. DYNAMIC MODEL

The quadrocopter is described by six degrees of freedom: the translational position $(x,\, y,\, z)$ is measured in the inertial coordinate system O as shown in Figure 1. The vehicle attitude V is defined by the rotation matrix ${}^{\mathrm{O}}_{\mathrm{V}}R$. The rotation matrix is defined such that, when multiplying a vector v in the vehicle coordinate V system with it, the same vector, described in the inertial coordinate system O, is obtained:

$$ {}^{\mathrm{O}}v = {}^{\mathrm{O}}_{\mathrm{V}}R\, {}^{\mathrm{V}}v \tag{1} $$

2.1 Control Inputs

The control inputs of the quadrotor vehicle are the desired rotational rates about the vehicle body axes, ω_x, ω_y, and ω_z, and the mass-normalized collective thrust, a, as shown in Figure 2.

High-bandwidth controllers on the vehicle track the desired rates using feedback from gyroscopes. The quadrocopter has very low rotational inertia, and can produce high torques due to the outward mounting of the propellers. This results in very high achievable rotational accelerations $\dot{\omega}_x$ and $\dot{\omega}_y$ on the order of $200\,\mathrm{rad/s^2}$. The vehicle has a very fast response time to changes in the desired rotational rate (experimental results have shown time constants on the order of 20ms for changes that do not saturate the motors). It is therefore assumed that we can directly control the vehicle body rates and ignore rotational acceleration dynamics. Rotational accelerations $\dot{\omega}_z$ are created by causing a drag difference between propellers rotating in opposite directions. Achievable accelerations are significantly lower at about $19\,\mathrm{rad/s^2}$. However, we will show that ω_z does not greatly influence the dynamics of the vehicle in this algorithm.

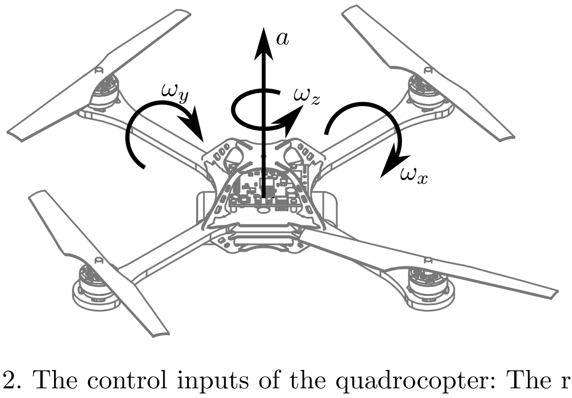

Fig. 2. The control inputs of the quadrocopter: The rotational rates ω_x, ω_y, and ω_z are tracked by an on-board controller, using gyroscope feedback.

Like the vehicle body rates, it is assumed that the thrust can be changed instantaneously. Experimental results have shown that the true thrust dynamics are about as fast as the rotational dynamics, with propeller spin-up being noticeably faster than spin-down.

It is further assumed that all control inputs are subject to saturation. The magnitude of the vehicle body rates are limited (such limitations can be caused, for example, by the range of the gyroscopes, or limitations of the body rate tracking controllers). The collective thrust is limited by a minimum and a maximum thrust

$$ a_{\min} \leq a \leq a_{\max}, \tag{2} $$

where $a_{\min} > 0$. This limitation is motivated by typical quadrotor vehicles having propellers of fixed-pitch type, and not being able to reverse their direction of rotation in flight.

2.2 Equations of Motion

The translational acceleration of the vehicle is dictated by the attitude of the vehicle and the total thrust produced by the four propellers. The translational acceleration in the inertial frame is

$$ \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = {}^{\mathrm{O}}_{\mathrm{V}}R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\mathrm{g} \end{bmatrix} . \tag{3} $$

The change of vehicle attitude is related to the rotational control inputs through [Hughes, 1986]

$$ {}^{\mathrm{O}}_{\mathrm{V}}\dot{R} = {}^{\mathrm{O}}_{\mathrm{V}}R \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} . \tag{4} $$

3. FEASIBILITY CONDITIONS FOR TRAJECTORIES

We calculate the control inputs for a given trajectory, allowing the inspection of the effects of the control input limitations on the feasibility of trajectories. Feasible trajectories must fulfill the equations of motion (3) and (4) and the required control inputs must not exceed allowable values.

3.1 Control Inputs for a Given Trajectory

Let $(x(t),\, y(t),\, z(t))$ denote a candidate vehicle trajectory. For notational convenience, we omit the time dependency from here on. Taking the second derivative of the trajectory and combining it with the translational equation of motion (3), we introduce the vector f representing the total mass-normalized forces required by the quadrotor to follow the trajectory:

$$ f := \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathrm{g} \end{bmatrix} = {}^{\mathrm{O}}_{\mathrm{V}}R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} . \tag{5} $$

Using the two-norm (denoted by $\|\cdot\|$), the thrust a required to follow the trajectory can be calculated:

$$\|f\| = \left\| {}^{\mathrm{O}}_{\mathrm{V}}R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} \right\| = a \ , \tag{6}$$

as the rotation matrix does not change the two-norm of a vector. The direction of thrust

$$\bar{f} := \frac{f}{\|f\|} \tag{7}$$

can be seen to define the third column of the rotation matrix by substituting $\bar{f}$ back into Equation (5):

$${}^{\mathrm{O}}_{\mathrm{V}}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \bar{f} \tag{8}$$

Taking the derivative of the above equation and combining it with Equation (4) gives two of the vehicle body rates as functions of the current attitude and $\dot{\bar{f}}$:

$$\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} = {}^{\mathrm{V}}_{\mathrm{O}}R \dot{\bar{f}} \ . \tag{9}$$

Equation (6) and the first two rows of Equation (9) provide three equations for four unknown control inputs. To determine the control input trajectories, a fourth equation must be additionally specified, which can be taken to be a user-defined constraint on ω_z ($\omega_z = 0$, for example).

3.2 Feasibility Conditions

Feasibility constraints for trajectories can now be calculated from the control input constraints:

Collective Thrust The collective thrust calculated from Equations (5) and (6) must lie between the minimum and maximum thrust, i.e.

$$a_{\min} \le \|f\| = \sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + \mathrm{g})^2} \le a_{\max} \ , \tag{10}$$

Rotational Rates The actual body rate control inputs ω_x and ω_y can only be computed if a constraint on ω_z is specified in addition to the trajectory. However, they can be bounded using the unit norm property of the rotation matrix [Bernstein, 2005] with Equation (9):

$$\omega_{x,y} \le \left\| \dot{\bar{f}} \right\| . \tag{11}$$

4. TRAJECTORY PLANNING

We seek to find a way to plan trajectories from any given vehicle position, velocity, and attitude to a given target position. We limit the planning to trajectories ending with the vehicle at rest. We seek trajectories that reach the target point as quickly as possible. Our approach is similar to the one presented for omni-directional ground robots in Purwin and D'Andrea [2006] and Sherback et al. [2006], but extended to the three-dimensional case and the dynamics of quadrotor vehicles.

The three degrees of freedom are decoupled, and time-optimal jerk trajectories are planned for each of them separately. The feasibility of the planned trajectories is then checked. If it is found to be infeasible, it is replanned with reduced jerk constraints, which eventually guarantees feasibility.

4.1 Decoupling of the Coordinates

To simplify the three-dimensional planning problem, we perform the planning separately for each of the coordinates x, y, and z. The three axes are coupled in the acceleration constraint (10) and the rotational control inputs (9). In order to allow decoupling, we introduce two simplifications:

Acceleration Constraint The acceleration constraint is decoupled by allocating a constant maximum allowable acceleration magnitude to each coordinate separately ($\ddot{x}_{\max}, \ddot{y}_{\max}, \ddot{z}_{\max}$). These maximum accelerations are chosen such that they fulfill constraint (10):

$$\sqrt{\ddot{x}^2_{\max} + \ddot{y}^2_{\max} + (\ddot{z}_{\max} + \mathrm{g})^2} \le a_{\max} \ , \tag{12}$$

$$\mathrm{g} - \ddot{z}_{\max} \ge a_{\min} \ . \tag{13}$$

Control Inputs In order to decouple the control inputs, we use the jerk of the three translational degrees of freedom as planning input

$$v := (\dddot{x},\ \dddot{y},\ \dddot{z}) \ , \tag{14}$$

which is related to the actual control inputs through (9) by explicitly calculating $\dot{\bar{f}}$:

$$\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} = {}^{\mathrm{V}}_{\mathrm{O}}R \left(\frac{v}{\|f\|} - \frac{f f^T v}{\|f\|^3} \right) \ . \tag{15}$$

It can be seen that the feasible values of the planning input v depend on f, making it difficult to choose allowable values a priori. Our approach is therefore to choose allowable magnitudes for each element of v, and to check the feasibility of the trajectory after solving the optimal control problem for all degrees of freedom.

4.2 Optimal Control Problem

Without loss of generality, we assume that the target position is the origin. As the planning is identical for all coordinates, we present it here on the example of the x-coordinate. Let $s = (s_1,\ s_2,\ s_3) = (x,\ \dot{x},\ \ddot{x})$ be the state. The time-optimal planning problem can then be stated as: Find the planning input u^* minimizing the final time t_f

$$u^* = \arg\min t_f \tag{16}$$

subject to the system dynamics

$$\dot{s}_1 = s_2 \tag{17}$$

$$\dot{s}_2 = s_3 \tag{18}$$

$$\dot{s}_3 = u \ , \tag{19}$$

the initial and final conditions

$$s(t = 0) = s_0 \tag{20}$$

$$s(t = t_f) = 0 \tag{21}$$

and the state and input constraints

$$|u| \le u_{\max} \tag{22}$$

$$|s_3| \le \ddot{x}_{\max} \ . \tag{23}$$

We will later iterate over $u_{\max}$ to give a feasible trajectory.

4.3 Necessary Optimality Conditions

We use Pontryagin's minimum principle (see, for example, Bertsekas [2005]) to derive necessary conditions for optimal input trajectories. The methodology we use to handle state constraints is the *direct adjoining approach* [Hartl

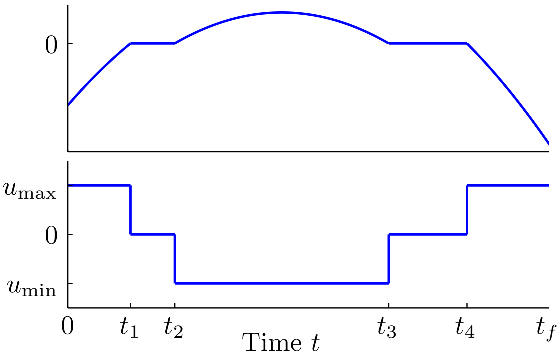

Fig. 3. Characteristic trajectory of the third adjoint variable λ_3 (top), and the optimal control input u^* (bottom).

et al., 1995], in which the augmented Hamiltonian function is defined by

$$\begin{aligned} H(s,u,\lambda,\eta) = {} & \lambda_1 s_2 + \lambda_2 s_3 + \lambda_3 u \\ & + \eta_1 (\ddot{x}_{\max} + s_3) + \eta_2 (\ddot{x}_{\max} - s_3) \ , \end{aligned} \tag{24}$$

where λ are the adjoint variables and η are state constraint multipliers that fulfill

$$\eta \geq 0 \tag{25}$$
$$\eta_1 = 0 \text{ if } s_3 > -\ddot{x}_{\max} \tag{26}$$
$$\eta_2 = 0 \text{ if } s_3 < \ddot{x}_{\max} \ . \tag{27}$$

The adjoint variables must fulfill

$$\dot{\lambda} = -\nabla_s H(s,u,\lambda,\eta), \tag{28}$$

which results in

$$\dot{\lambda}_1 = 0 \tag{29}$$
$$\dot{\lambda}_2 = \lambda_1 \tag{30}$$
$$\dot{\lambda}_3 = \lambda_2 + \eta_1 - \eta_2. \tag{31}$$

The optimal control u^* is the control input that minimizes the Hamiltonian function:

$$u^* = \arg\min H(s,u,\lambda,\eta) \tag{32}$$
$$= \arg\min \lambda_3 u. \tag{33}$$

Maurer [1977] proves in Corallary 5.2 (ii) that, for problems of this form, the adjoint variables λ are continuous. Furthermore, $\lambda_3 = 0$ must hold when a state constraint is active. The optimal control consists of *interior arcs* and *boundary arcs*. On interior arcs, $\eta_1 = \eta_2 = 0$, $\lambda_3 \neq 0$ and $u^* = \pm u_{\max}$. On boundary arcs, $s_3 = \pm\ddot{x}_{\max}$ and it follows that $\lambda_3 = 0$ and $u^* = 0$. The trajectory of λ_3 must be composed of parabolic shapes according to Equation (31), an example of which is shown in Figure 3 along with the corresponding control input.

It can be verified from the above constraints that the trajectory consists of at most five sections:

- $[0\ t_1]$ is an interior arc, with $u^* = \pm u_{\max}$.
- $[t_1\ t_2]$ is a boundary arc, with $u^* = 0$.
- $[t_2\ t_3]$ is an interior arc, with $u^* = \mp u_{\max}$.
- $[t_3\ t_4]$ is a boundary arc, with $u^* = 0$.
- $[t_4\ t_f]$ is an interior arc, with $u^* = \pm u_{\max}$.

4.4 Optimal Solutions

To fully specify the control trajectory, we must solve for the five times t_1, t_2, t_3, t_4, and t_f and the initial control input.

The solution is constrained by the three terminal state conditions (21). The equations of motion (17)–(19) are straightforward to integrate analytically, resulting in three equations of first, second, and third order in the switch times $t_1 \ldots t_f$, respectively.

Furthermore, each boundary arc induces one additional constraint, in that one of the following conditions must hold:

- If the duration of the boundary arc is nonzero ($t_2 - t_1 > 0$ or $t_4 - t_3 > 0$, respectively), then s_3 must be on the constraint at the beginning of the boundary arc ($|s_3(t_1)| = x_{3,\max}$ or $|s_3(t_3)| = x_{3,\max}$, respectively).
- If the above condition does not hold, the corresponding boundary arc must vanish ($t_2 - t_1 = 0$ or $t_4 - t_3 = 0$, respectively).

This yields five equations for five unknowns, meaning that the solution is fully determined by the constraints. The resulting system of equations is rather large and is therefore not shown here. It is available online on the first author's website at `www.idsc.ethz.ch/people/staff/hehn-m`. The solution to the system of equations is computed using a bisection root-finding algorithm.

4.5 Feasibility of Control Inputs

After the time-optimal trajectories have been calculated for each degree of freedom, the feasibility of the control inputs is checked. The thrust input a is guaranteed to be feasible because of the choice of $(\ddot{x}_{\max}, \ddot{y}_{\max}, \ddot{z}_{\max})$ according to Equations (12)-(13). For the rotaional control inputs, we calculate the maximum value of the bound (11) along the planned trajectory. Using Equation (15), the bound can be expressed as

$$\omega_{x,y} \leq \left\| \frac{v}{\|f\|} - \frac{f f^T v}{\|f\|^3} \right\|. \tag{34}$$

It is easy to bound $\omega_{x,y}$ for a planned trajectory using this equation: as v is piecewise constant and f is piecewise linear, the extrema of the above expression can easily be found by evaluating (34) at points where its derivative vanishes, and at the switching times $t_1 \ldots t_f$. The expressions for the extrema offer limited insight. They are therefore not reproduced in this paper, but may be inspected on the first author's website.

If the bound calculated exceeds the allowable value, the trajectory is replanned after reducing $u_{\max}$ for all degrees of freedom. By the triangle inequality and the Cauchy-Schwarz inequality [Bernstein, 2005], the expression (34) is at most

$$\omega_{x,y} \leq \frac{\|v\|}{\|f\|} + \frac{\|f f^T v\|}{\|f\|^3} \tag{35}$$

$$\omega_{x,y} \leq \frac{\|v\|}{\|f\|} + \frac{\|f\|^2 \|v\|}{\|f\|^3} \tag{36}$$

$$\omega_{x,y} \leq \|v\| \frac{2}{\|f\|}. \tag{37}$$

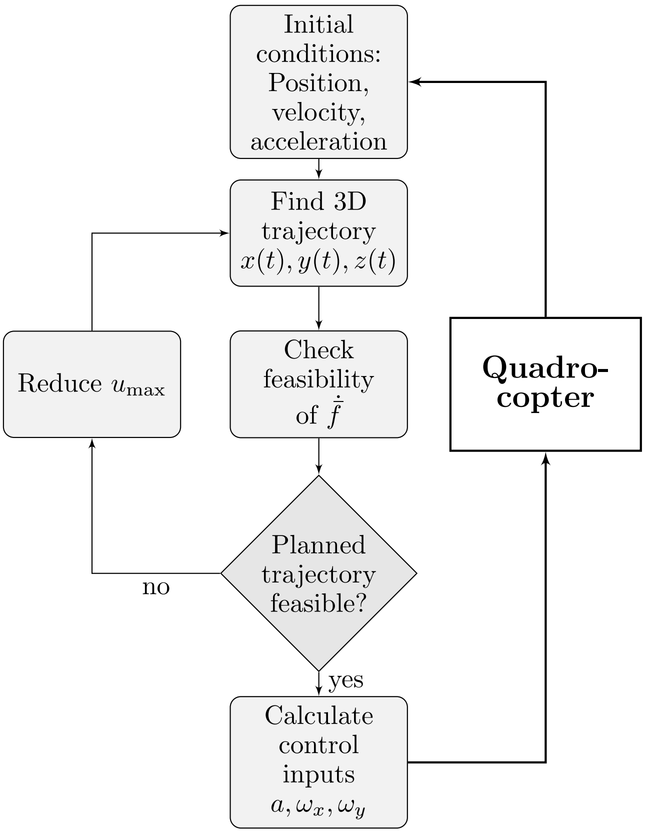
Initial conditions: Position, velocity, acceleration
Find 3D trajectory $x(t), y(t), z(t)$
Reduce $u_{\max}$
Check feasibility of $\dot{f}$
Quadro-copter
Planned trajectory feasible?
no
yes
Calculate control inputs a, ω_x, ω_y

Fig. 4. Closed-loop control using the trajectory planning. ω_z can be chosen freely (e.g. set to zero).

By the design of the trajectory, $\|f\|$ is bounded by equations (12)-(13). It can therefore be seen that this bound on $\omega_{x,y}$ scales with $\|v\|$. Reducing $u_{\max}$ for all degrees of freedom therefore guarantees the finding of a feasible trajectory.

5. CLOSED-LOOP CONTROL

The trajectory planning procedure is sufficiently fast to be performed for every controller update. We therefore close a feedback loop by re-planning the entire trajectory on each controller update, and applying the control inputs of the first control interval. The method is very similar to model predictive control (see, for example, García et al. [1989] for an overview) in that an optimal trajectory is generated at each time step.

However, the planning is carried out using the planning input v and not the true vehicle control inputs a, ω_x, ω_y, and ω_z. The control inputs must therefore be calculated from the planned trajectory.

Whereas the control inputs are usually assumed piecewise constant in model predictive control, the trajectory planned here allows for continuously varying control inputs (the planning input v is piecewise constant, but the control inputs ω_x, ω_y and a vary over time). Because actual commands to the underlying inner control loops are generated in discrete time intervals, we numerically average the control inputs over the (typically small) control interval, and command this average to the underlying control loops. This method has shown to work well in our experiments, however further investigation is necessary to determine the best discretization method.

As discussed in Section 3, the planned trajectory does not define ω_z. We choose to set $\omega_z = 0$ in order to minimize the control effort required for this degree of freedom. As the choice of ω_z is independent of the trajectory, it would also be possible to choose other values, such as those that maintain the vehicle heading constant. Numerically integrating Equation (4) over the control interval with the constraints (15) and ω_z allows the calculation of the body rates ω_x and ω_y for the interval.

The commands to the vehicle are completed by the collective thrust command. For this, no numerical integration is necessary as Equation (6) can be evaluated analytically.

Figure 4 shows the flow chart of closed-loop control using the trajectory planning. At every controller update, the initial conditions for the planning problem are updated. The initial position and velocity are states of the system, and the initial acceleration is constructed using the current vehicle attitude and the last thrust command using Equation (3). The path planning is then carried out and, if necessary, adapted to ensure feasibility of the trajectory. From the planned trajectory, the control inputs are calculated.

6. RESULTS

The planning and control algorithm presented herein has been implemented in the Flying Machine Arena, an indoor aerial vehicle development platform at ETH Zurich.

6.1 Experimental Setup

We use modified Ascending Technologies 'Hummingbird' quadrocopters [Gurdan et al., 2007] in our experiment. The vehicles are equipped with custom electronics, allowing greater control of the vehicle's response to control inputs, a higher dynamic range, and extended interfaces [Lupashin et al., 2010].

Commands are sent through a proprietary low-latency 2.4 GHz radio link at a frequency of 50 Hz. Command loss is in the range of 0.1%. An infrared motion tracking system provides precise vehicle position and attitude measurements at 200 Hz. The total closed-loop latency is approximately 30 ms.

The position and attitude measurements are filtered by a state observer providing full state information. Furthermore, the observer predicts the system state into the future to compensate for the closed-loop latency, such that controllers can be designed for the latency-free system.

The controller computation is carried out on a conventional desktop computer using an Intel Core2 Duo processor at 3.16 GHz. A full evaluation of the trajectory planning and control algorithm, as depicted in Figure 4, always took less than 0.2 ms (1% of the planning period) during these experiments.

6.2 Experimental Results

During experimental flights, target points were manually controlled through keystrokes on the controlling computer.

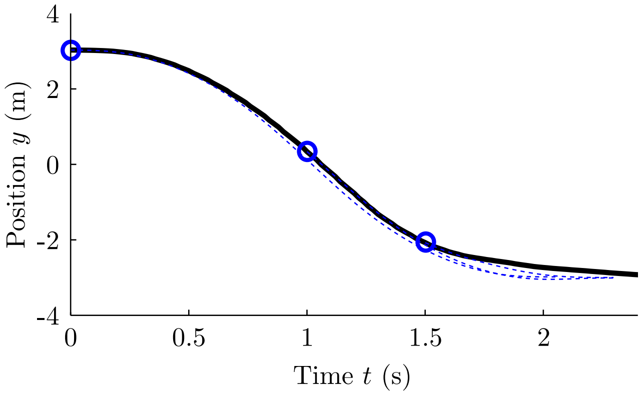

Fig. 5. One-dimensional flight trajectory during experiment. The vehicle starts at rest, and the target point is set to $y = -3\,\mathrm{m}$. The actual flight trajectory is depicted by the black line. Several trajectories planned along the way are shown by dotted blue lines, starting at the respective blue circle.

The maximum allowable accelerations were chosen to be $\ddot{x}_{\max} = \ddot{y}_{\max} = \ddot{z}_{\max} = 7\,\mathrm{m/s^2}$. The planning input limitation was set to $u_{\max} = 50\,\mathrm{m/s^3}$. A more conservative version of the feasibility check (34) was used [1] , the maximum permissible body rate command $\omega_{x,y}$ was set to 15 rad/s. A video showing the experiments presented here is available online at `www.idsc.ethz.ch/people/staff/hehn-m`.

Figure 5 shows a test of a translation of 6 meters along the the y-axis, with the set points for both other axes remaining constant. The test was started with the vehicle at rest. It can be seen that the vehicle followed the trajectory planned at the start of the experiment well. While re-planning occurs at the control update rate of 50 Hz, we only show a selection of planned trajectories.

Figure 6 shows the flight trajectory of a motion along both the x- and the y-axis. In this test, the vehicle was initially in motion, with the starting point of the trajectory representing the time when the target point was switched. The dynamic constraints lead to the rather smooth curve. An overshoot of about 0.2 m can be seen at the end of the maneuver.

Figure 7 shows the same experiment described above, with all coordinates visible. It is interesting to note the increasing error in z during the deceleration at the end of the trajectory. We suspect that strong aerodynamic effects caused this rise of the vehicle. The vehicle reached a peak speed of 7.1 m/s, a speed at which aerodynamic effects contribute significantly to the system dynamics. Huang et al. [2009] show that, at such speeds, the flight dynamics of quadrocopters begin resembling those of an airplane. Furthermore, the propellers can produce up to 70% more thrust for a given power.

Figure 8 shows the dynamic replanning capabilities in an extreme situation: The experiment starts with the vehicle at a velocity of 7.5 m/s, and the target point 8.5 m away from its current position. The total flight duration shown in the figure is 4 s. As seen before, an upwards deviation from the planned trajectory is noticeable when the vehicle decelerates, both at the beginning and the end of the trajectory. This matches the aerodynamic reasoning above. The effect has been observed consistently when decelerating from high speeds during the testing of the algorithm.

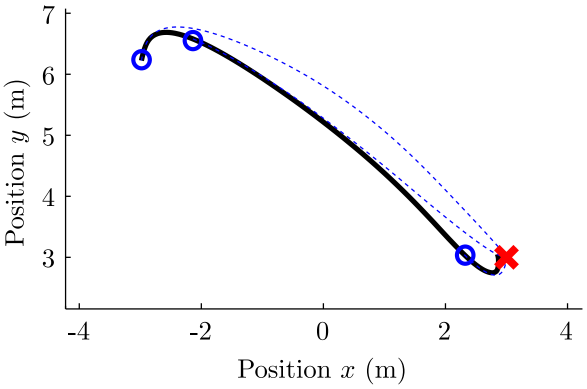

Fig. 6. Flight trajectory during experiment. The target point is marked by a red cross. The vehicle starts at nonzero velocity and acceleration. Its actual trajectory is shown in black, and a selection of planned trajectories are shown by dotted blue lines, starting at the respective blue circle.

7. CONCLUSION AND OUTLOOK

in this paper, we presented a planning algorithm that respects the dynamic and input constraints of quadrotor vehicles. The algorithm is fast enough to be evaluated at every controller update, and therefore allows feedback control based on planned trajectories. This planning and feedback control method shows promise, as validated by experiments in the Flying Machine Arena.

In the current algorithm, we have chosen to set fixed limitations on the accelerations allowed for each coordinate. One means of improving the planning algorithm would be to set variable acceleration bounds. This would permit

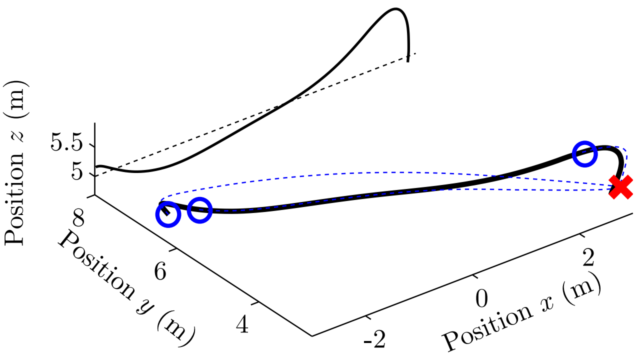

Fig. 7. 3D flight trajectory during experiment. The target point is marked by a red cross. The vehicle starts at nonzero velocity and acceleration. Its actual trajectory is shown in bold black, and the planned path at various instances in time in dotted blue (starting at the blue circles). The thin black line shows the projection of the actual trajectory on the x-z-plane.

[1] The experiments were carried out before the rotational bound results (34) were derived. Instead, a strictly more conservative bound, obtained by using the one-norm of Equation (9), was used.

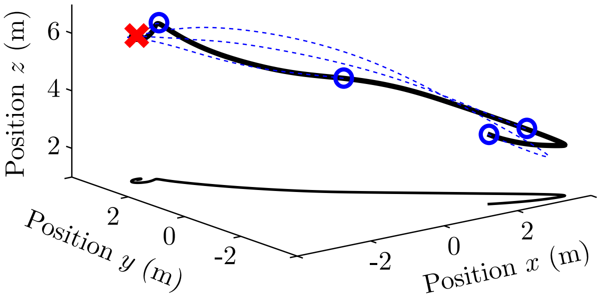

Fig. 8. Flight trajectory during 'extreme-case' experiment. The target point is marked by a red cross. The vehicle starts at high velocity and acceleration. Its actual trajectory is shown in bold black, and planned paths at various instances in time in blue (starting at the circles). The thin black line shows the projection of the actual trajectory on the x-y-plane.

us to determine the distribution of allowable accelerations that minimizes the total duration of the trajectory. As is, the algorithm is fast enough to evaluate approximately 100 different combinations of allowable accelerations for each controller update, making online parameter optimization a viable solution. Furthermore, synchronizing the arrival time of the three degrees of freedom would allow planning trajectories to states where the vehicle is not at rest.

A second improvement would be to consider the asymmetry of the quadrotor dynamics about the z-axis in the acceleration limits: During the entire planning procedure, a fixed amount of thrust is reserved for vertical accelerations. As gravity assists vertical decelerations, the reserved thrust is not required during deceleration phases and could be used to increase the speed of the trajectory in horizontal directions. This could be represented by time-varying allowable horizontal accelerations in the planning algorithm.

Finally, it should be possible to mitigate dominant aerodynamic effects. Experimental results showed systematic discrepancies between planned flight trajectories and actual trajectories. The largest effect was a significant rise of the vehicle when decelerating from high speeds. This effect can be explained by well-known aerodynamic effects of quadrotors moving at high speeds and high angles of attack. The error appears to be systematic and repeatable. It should therefore be possible to compensate for this effect, either by refining the dynamic model or through learning algorithms.

We are currently seeking to extend the planning algorithm with these improvements to further improve the flight performance. The implementation and experimental validation of the method as presented herein is an important step towards this end. The algorithm can be employed as-is, or built upon.

ACKNOWLEDGEMENTS

The authors wish to thank Sergei Lupashin for his innumerable contributions to the Flying Machine Arena hardware and software, Michael Sherback for his work on the quadrotor estimation and simulation algorithms, and Angela Schöllig for her work on the Flying Machine Arena testbed.

REFERENCES

Dennis S. Bernstein. *Matrix Mathematics.* Princeton University Press, 2005.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. I.* Athena Scientific, third edition, 2005.

Y. Bouktir, M. Haddad, and T. Chettibi. Trajectory Planning for a Quadrotor Helicopter. In *Mediterranean Conference on Control and Automation.* Ieee, June 2008.

Ian D. Cowling, Oleg A. Yakimenko, and James F. Whidborne. A Prototype of an Autonomous Controller for a Quadrotor UAV. In *European Control Conference*, 2007.

Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice - A survey. *Automatica*, 25(3):335–348, May 1989.

Daniel Gurdan, Jan Stumpf, Michael Achtelik, Klaus-Michael Doth, Gerd Hirzinger, and Daniela Rus. Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz. *IEEE International Conference on Robotics and Automation*, April 2007.

Richard F. Hartl, Suresh P. Sethi, and Raymond G. Vickson. A Survey of the Maximum Principles for Optimal Control Problems with State Constraints. *SIAM Review*, 37(2):181–218, 1995.

Gabriel M. Hoffmann, Steven L. Waslander, and Claire J. Tomlin. Quadrotor Helicopter Trajectory Tracking Control. In *IEEE Conference on Decision and Control*, 2008.

Jonathan P. How, Brett Bethke, Adrian Frank, Daniel Dale, and John Vian. Real-Time Indoor Autonomous Vehicle Test Environment. *IEEE Control Systems Magazine*, 28(2):51–64, 2008.

Haomiao Huang, Gabriel M. Hoffmann, Steven L. Waslander, and Claire J. Tomlin. Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering. *IEEE International Conference on Robotics and Automation*, May 2009.

Peter C. Hughes. *Spacecraft Attitude Dynamics.* John Wiley & Sons, 1986.

Li-Chun Lai, Chi-Ching Yang, and Chia-Ju Wu. Time-Optimal Control of a Hovering Quad-Rotor Helicopter. *Journal of Intelligent and Robotic Systems*, 45(2):115–135, June 2006.

Sergei Lupashin, Angela Schöllig, Michael Sherback, and Raffaello D'Andrea. A Simple Learning Strategy for High-Speed Quadrocopter Multi-Flips. In *IEEE International Conference on Robotics and Automation*, 2010.

H. Maurer. On Optimal Control Problems with Bounded State Variables and Control Appearing Linearly. *SIAM Journal Control and Optimization*, 15(3):345–362, 1977.

Oliver Purwin and Raffaello D'Andrea. Trajectory generation and control for four wheeled omnidirectional vehicles. *Robotics and Autonomous Systems*, 54:13–22, 2006.

Michael Sherback, Oliver Purwin, and Raffaello D'Andrea. Real-Time Motion Planning and Control in the 2005 Cornell RoboCup System. *Lecture Notes in Control and Information Sciences*, 335:245–264, 2006.