Real-Time Trajectory Generation for Interception Maneuvers with Quadrocopters

Markus Hehn and Raffaello D'Andrea

Abstract— **This paper presents an algorithm that permits the calculation of interception maneuvers for quadrocopters. The translational degrees of freedom of the quadrocopter are decoupled. Pontryagin's minimum principle is used to show that the interception maneuver that minimizes the time to rest after the interception is identical to the time-optimal maneuver that drives the vehicle to the position at which it comes to rest after the interception. This fact is leveraged to apply previously developed, computationally efficient methods for the computation of interception maneuvers. The resulting trajectory generation algorithm is computationally lightweight, permitting its use as an implicit feedback law by replanning the trajectory at each controller update. The validity and performance of the approach is demonstrated experimentally by intercepting balls mid-flight. The real-time trajectory generation permits to take into account changes in the predicted ball flight path at each controller update.**

I. Introduction

Quadrocopters have been widely adopted as experimental platforms for research in flying robotics (see, for example, the testbeds [15], [20]). Reasons for the popularity of these vehicles include the ability to hover, mechanical simplicity and robustness, and their exceptional maneuverability due to typically high thrust-to-weight ratios combined with the off-center mounting of the propellers.

From a controls perspective, a recent focus has been the planning and following of trajectories that exploit the dynamical capabilities of these vehicles. Results include algorithms that plan trajectories from classes of motion primitives, such as lines [14] or polynomials [4], [7], while others solve an optimal control problem for approximate or full vehicle dynamics (e.g. for minimum snap [19] or minimum time [12]).

In this paper, we consider the problem of using a quadrocopter for the purpose of interception. The general interception problem has been studied in an optimal control context for several decades (see, for example, [5] and references therein). Variations of the problem have also been studied in robotics (e.g. for robotic arms [1] and ground robots [22]) For quadrotor applications, interception problems have been treated in a number of scenarios, including ball juggling, where the interception was more strongly constrained to occur at a specified velocity and attitude [21]. In [3], a ball flight path was intercepted on a given plane by setting the controller reference position to the interception point.

The method we present herein permits the computation of interception maneuvers for quadrocopters in real time. These maneuvers are optimal in that they minimize the time to rest after the interception event, when decoupled dynamics are assumed. This paper shows that the resulting maneuver structure is identical to the time-optimal maneuver that brings the vehicle to the position at which it comes to rest after the interception. This allows us to apply the efficient trajectory planning algorithm that was introduced in [11]. Because the entire trajectory (consisting of the interception and to-rest motion thereafter) is planned, it is easy to verify additional constraints such as, for example, maximum allowable positions.

The trajectory generation algorithm is computationally light weight, permitting us to recompute trajectories in real time at update rates on the order of tens to hundreds of replannings per second. This also permits us to use the trajectory generation as an implicit feedback law by applying the control inputs of the first section of the planned trajectory (similar to model predictive control [8]) at each controller update. Furthermore, trajectories can be planned from arbitrary initial states, and the generated trajectory is guaranteed to be feasible with respect to the dynamic and input constraints of the vehicle.

The remainder of this paper is structured as follows: In Section II, we introduce the dynamic model of the quadrocopter used in the trajectory generation. In Section III, we provide a brief overview of the previously presented trajectory generation algorithm for time-optimal maneuvers to a specified position. Section IV presents the interception problem and includes the derivation of optimality conditions for interception trajectories. In Section V, we discuss the implementation of an algorithm computing such interception maneuvers. Section VI describes the experimental setup we use to intercept balls mid-flight and presents experimental results, while Section VII draws conclusions, highlighting directions for future research.

II. VEHICLE DYNAMICS

The quadrocopter is described by six degrees of freedom: Its translational position $(x,\ y,\ z)$ in the inertial frame O and its attitude V, defined by the rotation matrix ${}^{\mathrm{O}}_{\mathrm{V}}R$.

The four control inputs of the 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 1.

It is assumed that the three rotational rates ω_x, ω_y, ω_z can be changed arbitrarily fast. This is motivated by the large rotational accelerations quadrocopters can achieve due to their ability to produce high torques and their low rotational

This work is in part funded by the Swiss National Science Foundation.

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

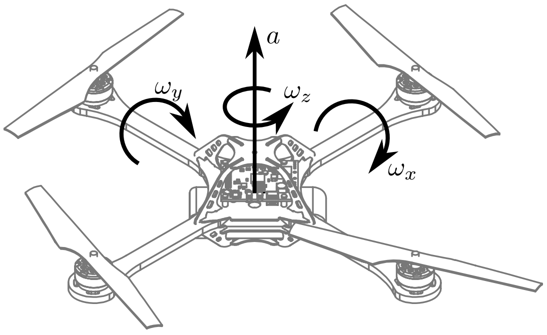

Fig. 1. The four control inputs of the quadrotor vehicle. The rotational rates ω_x, ω_y, and ω_z are assumed to follow commands without dynamics or delay. This is motivated by a high-bandwidth on-board controller.

inertia [17], which allow the rotational rate commands to be tracked with very high bandwidth on board the vehicle.

Analogously to the vehicle body rates, we assume that the thrust can be changed instantaneously. Experimental results have shown that the true thrust dynamics, caused by the dynamics of the motors changing speed, are about as fast as the rotational rate dynamics.

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

$$a_{\min} \leq a \leq a_{\max} \ , \tag{1}$$

where $a_{\min} > 0$. This positive lower bound is motivated by the fact that typical quadrotor vehicles have propellers of fixed-pitch type, and are not able to stop or reverse the propellers' direction of rotation in flight.

A. Equations of Motion

The translational acceleration of the vehicle is dictated by its attitude and the collective thrust control input. In the inertial frame, the translational acceleration is

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = {}^{\mathrm{o}}_{\mathrm{v}}R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\mathrm{g} \end{bmatrix} \ , \tag{2}$$

where g denotes the gravitational acceleration.

The change of vehicle attitude is related to the rotational control inputs through [16]

$$ {}^{\mathrm{o}}_{\mathrm{v}}\dot{R} = {}^{\mathrm{o}}_{\mathrm{v}}R \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \ . \tag{3}$$

III. TRAJECTORY GENERATION ALGORITHM

In this section, we provide an overview of the method used to generate trajectories from arbitrary initial conditions to a target point. This approach was introduced in [11], and is described here briefly because it will later be used.

The trajectory generation problem is simplified by approximating the quadrotor dynamics with three triple integrators:

$$\begin{bmatrix} \dddot{x} \\ \dddot{y} \\ \dddot{z} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \ , \tag{4}$$

and planning the trajectory in the jerks $(v_x(t), v_y(t), v_z(t))$. The control inputs along a trajectory $(x(t), y(t), z(t))$ are

$$f(t) := \begin{bmatrix} \ddot{x}(t) \\ \ddot{y}(t) \\ \ddot{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathrm{g} \end{bmatrix} \ , \tag{5}$$

$$\bar{f}(t) := \frac{f(t)}{\|f(t)\|} \ , \tag{6}$$

$$a(t) = \|f(t)\| \ , \tag{7}$$

$$\begin{bmatrix} \omega_y(t) \\ -\omega_x(t) \\ 0 \end{bmatrix} = {}^{\mathrm{v}}_{\mathrm{o}}R(t)\dot{\bar{f}}(t) \ . \tag{8}$$

For a trajectory to be feasible, the resulting control inputs a, ω_x, and ω_y must lie within their allowable sets. In order to satisfy the thrust constraint (1), constant acceleration bounds for each degree of freedom are introduced. Furthermore, it was shown that it is difficult to choose allowable jerk values such that the rotational rate control inputs (8) do not exceed the vehicle body rate limitations. It is however straightforward to compute the rotational rates along a planned trajectory. If these exceed limitations, it was shown that a feasible trajectory can always be found by reducing the allowable jerk values.

A trajectory to the target position can be computed for each degree of freedom of the approximate dynamics (4), with given acceleration and jerk constraints. Time-optimal trajectories from arbitrary initial conditions to a target position (to be reached at rest) were presented. Using Pontryagin's minimum principle, it follows that the switching function is of parabolic shape with additional intervals at the zero crossings in which it remains zero. The optimal control input v^* is bang-singular, consisting of at most five distinct regions:

- $[0 \, t_1)$: $v^* = \pm v_{\max}$,
- $[t_1 \, t_2)$: $v^* = 0$,
- $[t_2 \, t_3)$: $v^* = \mp v_{\max}$,
- $[t_3 \, t_4)$: $v^* = 0$,
- $[t_4 \, t_f]$: $v^* = \pm v_{\max}$,

where $v_{\max}$ denotes the maximum allowable jerk. In this solution, the intervals $[t_1 \, t_2)$ and $[t_3 \, t_4)$ represent singular arcs, in which the control input is determined by the acceleration remaining constant on its boundary. The initial control input and the five times $t_1 \ldots t_f$ fully define the maneuver.

The performance of this trajectory generation algorithm has been demonstrated in a number of experiments [11]. It was shown that it can be used as an implicit feedback law by re-planning a trajectory for each controller update. The computational load caused by this is minor, with computation taking no longer than $0.2\,\mathrm{ms}$ on a conventional desktop computer.

We will now adapt the trajectory generation algorithm, not to reach a target position as quickly as possible, but to cross a specified position at a specified time.

IV. THE INTERCEPTION MANEUVER

While the time-optimal to-rest maneuvers presented above are very useful in many applications, situations may arise in which a specified position must be reached more quickly than it is possible to reach at rest. Using the same coordinate decoupling (4) as before, we will now derive trajectories for a given *interception point* $(\hat{x}, \hat{y}, \hat{z})$, which must be reached at a specified *interception time* $\hat{t}$.

Because the interception constraint is not sufficient to uniquely define the trajectory, we further require the planned trajectories to bring the vehicle to rest as quickly as possible after the interception. This choice provides two advantages:

- The problem statement now includes not only the motion to intercept the position at the right time, but also the motion to bring the vehicle back to rest after the interception. This is desirable because it permits easy verification of constraints such as a maximum desirable displacement.
- The choice of the time to rest as a cost function forces aggressive deceleration after the time of interception, avoiding excessive overshoot.

We will now formally state the trajectory generation problem described above, allowing the optimality conditions to be applied in order to find solutions to it.

A. Problem statement

We describe the problem for a single degree of freedom, assuming that we decouple the dynamics according to equation (4). We denote the degree of freedom q, and state the optimal control problem as follows: Let $s = (s_1,\ s_2,\ s_3) = (q,\ \dot{q},\ \ddot{q})$ be the state. The objective is to minimize t_f subject to the system dynamics

$$\dot{s}_1 = s_2\ , \tag{9}$$

$$\dot{s}_2 = s_3\ , \tag{10}$$

$$\dot{s}_3 = v\ , \tag{11}$$

and the initial, interception, and final state constraints

$$s(t=0) = s_0\ , \tag{12}$$

$$s_1(t=\hat{t}) = \hat{q}\ , \tag{13}$$

$$s_2(t=t_f) = s_3(t=t_f) = 0\ , \tag{14}$$

where $\hat{t}$ is the interception time and $\hat{q}$ is the interception position. Furthermore, the input and state constraints

$$|v| \leq v_{\max}\ , \tag{15}$$

$$\ddot{q}_{\min} \leq s_3 \leq \ddot{q}_{\max} \tag{16}$$

must be satisfied.

We begin by noting that the algorithm presented in Section III already solves the above problem if it is possible to reach $\hat{q}$ such that t_f is smaller than $\hat{t}$: Because the computed trajectory reaches $\hat{q}$ at rest, the interception constraint (13) will be satisfied by this time-optimal motion followed by the vehicle remaining at rest until the interception time (i.e. $v^* = 0$ for $t_f < t \leq \hat{t}$). From here on, we will assume that $\hat{t}$ is smaller than t_f, and will now derive the optimality conditions and resulting structure of optimal maneuvers.

B. Necessary Optimality Conditions

Analogous to the derivations for the original trajectory generation algorithm, we apply the minimum principle (see, for example, [2]) to derive necessary conditions for optimal input trajectories. The state constraints (16) are handled using a direct adjoining approach [10], in which the Hamiltonian function is augmented by the state constraints. With the cost given to be the final time, the Hamiltonian is then

$$\begin{aligned} H(s, v, \lambda, \eta) = 1 &+ \lambda_1 s_2 + \lambda_2 s_3 + \lambda_3 v \\ &+ \eta_1 \left(-\ddot{q}_{\min} + s_3\right) + \eta_2 \left(\ddot{q}_{\max} - s_3\right)\ , \end{aligned} \tag{17}$$

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

$$\eta\ \geq 0\ , \tag{18}$$

$$\eta_1 = 0 \text{ if } s_3 > \ddot{q}_{\min}\ , \tag{19}$$

$$\eta_2 = 0 \text{ if } s_3 < \ddot{q}_{\max}\ . \tag{20}$$

The adjoint variables evolve over time according to

$$\dot{\lambda} = -\nabla_s H(s, v, \lambda, \eta), \tag{21}$$

from which it follows that

$$\dot{\lambda}_1 = 0\ , \tag{22}$$

$$\dot{\lambda}_2 = \lambda_1\ , \tag{23}$$

$$\dot{\lambda}_3 = \lambda_2 + \eta_1 - \eta_2\ . \tag{24}$$

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

$$v^* = \arg\min H(s, v, \lambda, \eta) = \arg\min \lambda_3 v\ . \tag{25}$$

At the interception time $\hat{t}$, the discontinuity in the first adjoint variable [5] is

$$\lambda_1(\hat{t}^-) = \lambda_1(\hat{t}^+) + \nu\ , \tag{26}$$

where $\hat{t}^-$ and $\hat{t}^+$ signify just before and just after $\hat{t}$, respectively, and ν is a constant Lagrange multiplier. The other two adjoint variables λ_2, λ_3 remain continuous over the interception time. Because the final state $s_1(t_f)$ is free, the costate constraint at the final time is [2]

$$\lambda_1(t_f) = 0\ . \tag{27}$$

It was shown in [18] that, for problems of this form, the adjoint variables λ are continuous when the acceleration constraint (16) becomes active or stops being active. Furthermore, $\lambda_3 = 0$ must hold over the duration in which a state constraint is active.

The costate dynamics (22)-(24) form a double integrator with respect to λ_3. With λ_1 being zero from $\hat{t}^+$ onwards, and with the intersection time condition (26), it follows that the trajectory of λ_3 has the following shape, which is sketched in Figure 2:

- In the time interval $\left[0\ \hat{t}^-\right]$, λ_3 has a parabolic shape because $\ddot{\lambda}_3 = \lambda_1$ is constant when no constraints are active.
- In the time interval $\left[\hat{t}^+\ t_f\right]$, $\ddot{\lambda}_3 = \lambda_1 = 0$, implying that λ_3 has constant slope when no constraints are active.

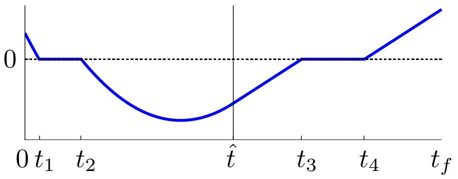

Fig. 2. Sketch of the adjoint variable trajectory λ_3 for an example maneuver. At the interception time $\hat{t}$, the shape switches from parabolic to constant slope as λ_1 jumps to zero. The intervals $[t_1\ t_2)$ and $[t_3\ t_4)$ represent singular arcs, in which one of the acceleration constraints (16) is active.

- Whenever λ_3 has a zero crossing, an acceleration constraint may become active, implying that λ_3 remains zero while the constraint is active.

C. *Equivalence to time-optimal motions*

It follows from the above constraints that the maneuver minimizing the to-rest duration with the interception constraint must have the same structure as the time-optimal trajectories presented in the previous section: Both consist of, at most, three regular arcs (where $\lambda_3 \neq 0$) and two singular arcs (where $\lambda_3 = 0$). Both trajectories are fully specified by the five times t_1, t_2, t_3, t_4, and t_f and the initial control input. For the interception trajectory, an additional constraint arises because the switching function λ_3 is linear after the interception time: The remaining trajectory can only contain two regular arcs and one singular arc.

Now, assume that we have computed an interception maneuver that satisfies the above optimality conditions, implying that it minimizes the to-rest duration after the interception. The maneuver terminates at some position q_f (by definition of the maneuver this position is reached at rest). Then this maneuver is identical to the time-optimal maneuver from the same initial conditions to the position q_f as described in Section III: Because both the time-optimal maneuver to q_f and the interception maneuver are fully defined by the switching times and the initial control input, and all boundary constraints are satisfied for both maneuvers, this must be the case.

Conversely, assume that we have computed a time-optimal maneuver to the position q_f. Then for all intermediate positions occurring after the time t_2, this is the interception maneuver that is optimal with respect to the above optimality conditions (we must limit the intermediate positions to ones occurring after t_2 due to the constant slope of λ_3 permitting only one zero crossing after $\hat{t}$).

The above equivalence of interception maneuvers and time-optimal maneuvers allows us to leverage the trajectory generation algorithm that we have developed for time-optimal maneuvers, as will be seen in Section V.

D. *Existence of solutions*

While the optimality conditions describe the structure of interception maneuvers that minimize the to-rest time after interception, it remains to verify that a maneuver satisfying the interception constraint (13) exists. Clearly, this will not always be the case: For small values of $\hat{t}$, the available control effort will not suffice to drive the system to $\hat{q}$ in time if $\hat{q}$ differs significantly from the motion dictated by the initial conditions. This is a fundamental difference to the motion to a given end point discussed in Section III, where all target points could be reached.

The existence of solutions can be formalized as the position $\hat{q}$ (with arbitrary velocity and acceleration) lying in the reachable set at time $\hat{t}$ for the given initial conditions. Using the fact that the reachable set is convex [6], it can be shown that the positions reachable at time $\hat{t}$ are bounded by the two trajectories for which $t_2 = \hat{t}$ (these are trajectories that apply the maximum or minimum possible control effort for the entire duration up to $\hat{t}$). The upper bound is reached by applying $v^* = v_{\max}$ in the interval $[0\ t_1)$, and the lower bound is reached by applying $v^* = -v_{\max}$.

Note that if no solution to the interception problem is found with this strategy, then there exists no other control input that can reach the interception point at time $\hat{t}$ [13]. This implies that, if we assume the decoupled dynamics (4), the strategy presented above provides an interception trajectory whenever the interception constraint (13) can be satisfied by the system dynamics.

V. COMPUTATION AND VERIFICATION OF INTERCEPTION MANEUVERS

In this section, we discuss how the properties of optimal interception trajectories can be used to efficiently compute interception maneuvers.

A. *Computation of maneuvers*

The identical structure of the interception maneuver and the time-optimal maneuver (recapitulated in Section III) permits us to compute maneuvers satisfying the interception constraint in a fashion that is very similar to the one employed for time-optimal maneuvers (see [11]): There is generally no closed-form solution for the five times $t_1 \ldots t_f$ from the constraint equations (12)-(14) and the singular arc constraints (19)-(20). It is however straightforward to find solutions using a one-dimensional bisection algorithm. While the computation of a time-optimal maneuver is carried out by iterating over the position at the end of the maneuver until the final position constraint is satisfied, we now iterate over the position at the interception time $\hat{t}$ in order to satisfy the interception constraint.

B. *Extremal points of the trajectory*

In many interception scenarios, it is important to not only satisfy the interception constraint, but to also come to rest within certain space constraints. We have not included such position constraints in the derivations in Section IV (the trajectory is already fully defined by the chosen constraints and optimality conditions, and additional constraints would complicate the trajectory structure significantly).

However, verifying such constraints once the trajectory has been planned is straightforward: The solution computed by the interception strategy presented herein results in a position trajectory that is piecewise polynomial of at most order three. This makes it simple to compute extremal points by finding points where the velocity vanishes. Such extremal points can then be compared to space restrictions.

C. Control effort distribution between the degrees of freedom

All previous derivations were based on a single degree of freedom. In order to intercept a position in 3D, all three degrees of freedom must be able to reach their respective target position by the interception time. To control each degree of freedom's ability to reach the target point in time, the acceleration constraints (16) can be varied. The three acceleration constraints are linked through condition (1). We parameterize the acceleration constraints as follows:

$$\ddot{z}_{\max} = c_z a_{\max} - \mathrm{g} \ , \tag{28}$$

$$\ddot{z}_{\min} = a_{\min} - \mathrm{g} \ , \tag{29}$$

$$\ddot{x}_{\max} = -\ddot{x}_{\min} = c_x \sqrt{a_{\max}^2 - (\ddot{z}_{\max} + \mathrm{g})^2} \ , \tag{30}$$

$$\ddot{y}_{\max} = -\ddot{y}_{\min} = \sqrt{1 - c_x^2}\sqrt{a_{\max}^2 - (\ddot{z}_{\max} + \mathrm{g})^2} \ , \tag{31}$$

where c_x and c_z are parameters. It is straightforward to verify that for all values of $c_x \in [0 \ \ 1]$, $c_z \in [\mathrm{g}/a_{\max} \ \ 1]$, the acceleration constraint (1) is satisfied.

It then remains to find parameters c_x and c_z such that each degree of freedom is able to reach the interception point. We propose a two-step strategy:

1. Starting from $c_z = 1$, find the lowest value c_z for which the vertical degree of freedom is able to both satisfy the interception constraint and not violate space restrictions, and then
2. vary c_x until both horizontal degrees of freedom reach the interception point and remain within space constraints.

The above method finds a three-dimensional interception trajectory that satisfies the interception constraint and possible space restrictions. It may however be beneficial to additionally specify a performance measure for the choice of c_x and c_z. We intend to investigate this in further research.

The above steps complete the description of an algorithm permitting the computation of a three-dimensional interception maneuver that is feasible with respect to the dynamics and constraints of the quadrocopter.

VI. EXPERIMENTAL RESULTS

The interception strategy presented herein has been tested in the Flying Machine Arena, an aerial vehicle development platform at ETH Zurich [17]. To demonstrate its performance, we apply it to the problem of "hitting" a ball mid-flight with a quadrotor vehicle.

A. Experimental setup

We use modified Ascending Technologies 'Hummingbird' quadrocopters [9]. The vehicles are equipped with custom electronics, allowing greater control of the vehicle's response to control inputs, sensors providing a higher dynamic range, and extended interfaces [17].

The trajectory generation algorithm is run on an off-board desktop computer at a rate of 70 Hz. For each computed trajectory, a command consisting of the three vehicle body rates and the collective thrust is sent to the vehicle through a low-latency 2.4 GHz radio link. The full state information (required for the initial conditions of the trajectory generation) is obtained from a state observer. The state observer receives precise vehicle position and attitude measurements from an infrared motion capture system at a rate of 200 Hz.

The ball that the vehicle is to hit is also tracked by the motion capture system, and its state is estimated using a Kalman filter combined with a drag coefficient estimator [21].

B. Determination of ball interception time

The interception algorithm presented above requires the interception time and position as inputs. To intercept the flight path of the ball, the interception time remains to be chosen. It is fixed as follows:

We consider the ball to be flying as soon as it crosses a threshold height. At this point, its flight path is predicted forward based on a first-principles model of its dynamics. The prediction is evaluated in discrete time steps from the current time up to the time at which the ball will fall below a certain height. The interception feasibility is verified for each of the discrete prediction points, with the additional constraint that the entire maneuver must remain within a specified volume. If the ball is interceptable, a range of possible interception times is found, the mean of which is chosen as the interception time to which the planning will occur from this point on. Figure 3 shows a ball being found, and a number of planned candidate interception trajectories. The interception time chosen by the algorithm lies in the middle of the candidate trajectories.

Note that further investigation is required to find the best way to choose the interception time. The method presented here was seen to work well in experiments, but its properties have not been analyzed thoroughly. We intend to investigate this in future work.

C. Flight results

Flight tests were carried out with the ball being thrown in varying directions. It was found that the vehicle was able

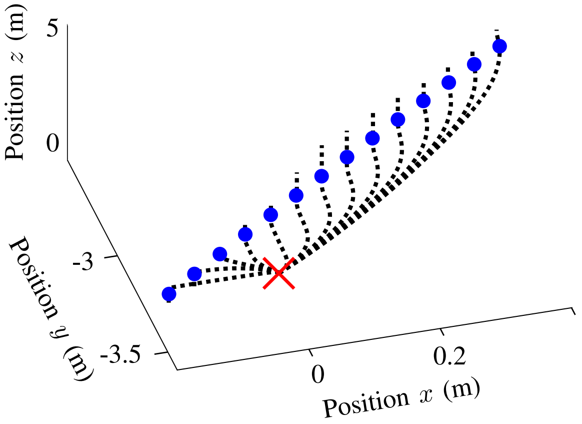

Fig. 3. A set of candidate trajectories planned when a ball is first detected. The red cross denotes the initial position of the vehicle. The blue dots are the ball positions predicted at various instances in time. The dotted black lines show the planned flight path for each of the predicted ball positions. The chosen interception time lies in the middle of these points.

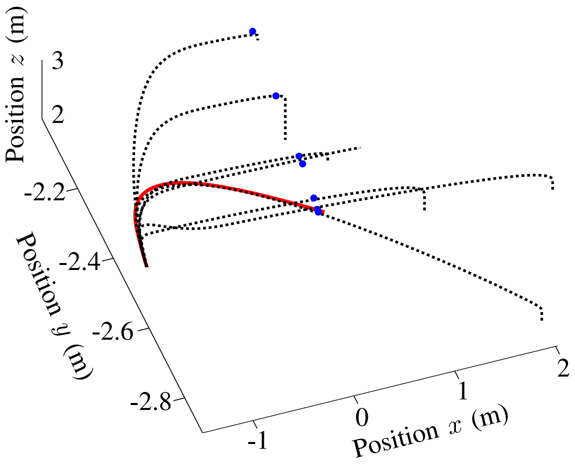

Fig. 4. Trajectories planned as the ball state and drag estimates converge. A selection of the planned trajectories is shown here (dotted black lines). The predicted interception location (blue points) changes by about $0.55\,\mathrm{m}$. The updated interception location is accounted for at every re-planning of the trajectory. The solid red line shows the trajectory that was actually flown. The vehicle intercepted the ball at the end of the red line.

to intercept ping-pong sized balls reliably when a feasible interception trajectory was found. A video showing a number of experiments may be found on the first authors web site.

Figure 4 shows an interception maneuver that highlights the advantage of the real-time capability of the trajectory generation algorithm: At the beginning of the maneuver, the ball state and drag estimates are subject to significant variation. Because the interception trajectory is replanned at every controller update, the moving interception point is naturally considered for each controller update. In this specific example, the interception point changes by about $55\,\mathrm{cm}$ over the course of $0.6\,\mathrm{s}$ before converging, and the vehicle successfully intercepted the ball.

VII. CONCLUSION & OUTLOOK

This paper introduced an interception trajectory generation algorithm for quadrotor vehicles. It was shown that the problem can be reformulated such that its structure is identical to the trajectory generation for time-optimal trajectories. For this class of trajectory generation problems, an efficient computational method has been developed previously and could be readily adapted. The successful implementation of this interception strategy has been verified experimentally by intercepting balls mid-flight.

In the trajectory design approach presented herein, a number of design parameters remained to be chosen, for example the control effort tradeoff between the three decoupled degrees of freedom. We are planning to investigate ways to systematically choose these such that the best possible performance is achieved.

Furthermore, we intend to investigate strategies for determining the optimal interception time for objects flying along trajectories, where optimality could be defined by criteria such as minimal vehicle velocity at interception time, the shortest time to interception, or other objectives.

An interesting extension of this work would be to include multiple interception constraints. Because the current interception method is computationally very lightweight, we believe that it should be possible to plan such multiple interception trajectories while maintaining real-time capabilities.

ACKNOWLEDGMENTS

The authors wish to thank Sergei Lupashin, Mark Müller, Angela Schöllig, and Michael Sherback for their contributions to the Flying Machine Arena testbed.

REFERENCES

[1] Berthold Bäuml, Thomas Wimböck, and Gerd Hirzinger. Kinematically Optimal Catching a Flying Ball with a Hand-Arm-System. In *International Conference on Intelligent Robots and Systems*, 2010.

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

[3] Patrick Bouffard, Anil Aswani, and Claire J Tomlin. Learning-Based Model Predictive Control on a Quadrotor: Onboard Implementation and Experimental Results. In *International Conference on Robotics and Automation*, 2012.

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

[5] Arthur Earl Bryson and Yu-Chi Ho. *Applied Optimal Control*. Taylor & Francis, 1975.

[6] S Chang. Minimal Time Control With Multiple Saturation Limits. *IEEE Transactions on Automatic Control*, 8(1):35–42, January 1963.

[7] 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.

[8] Carlos E. García, David M. Prett, and Manfred Morari. Model Predictive Control: Theory and Practice - A Survey. *Automatica*, 25(3):335–348, May 1989.

[9] 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. In *International Conference on Robotics and Automation*, 2007.

[10] 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.

[11] Markus Hehn and Raffaello D'Andrea. Quadrocopter Trajectory Generation and Control. In *IFAC World Congress*, 2011.

[12] Markus Hehn, Robin Ritz, and Raffaello D'Andrea. Performance Benchmarking of Quadrotor Systems Using Time-Optimal Control. *Autonomous Robots*, March 2012.

[13] Henry Hermes and Joseph P Lasalle. Functional Analysis and Time Optimal Control. *Mathematics in Science and Engineering*, 56, 1969.

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

[15] 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.

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

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

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

[19] D. Mellinger, N. Michael, and V. Kumar. Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. *The International Journal of Robotics Research*, January 2012.

[20] Nathan Michael, Daniel Mellinger, Quentin Lindsey, and Vijay Kumar. The GRASP Multiple Micro UAV Testbed. *Robotics Automation Magazine*, 17(3):56–65, 2010.

[21] Mark Muller, Sergei Lupashin, and Raffaello D'Andrea. Quadrocopter Ball Juggling. In *International Conference on Intelligent Robots and Systems*, 2011.

[22] Frieder Stolzenburg, Oliver Obst, and Jan Murray. Qualitative Velocity and Ball Interception. *Lecture Notes in Computer Science - KI2002: Advances in Artificial Intelligence*, 2479:95–99, 2002.