2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Time-optimal quadrotor flight*

Wannes Van Loock, Goele Pipeleers and Jan Swevers

***Abstract*— Solving the time-optimal path planning problem, where a system is brought from an initial to terminal state in minimal time while obeying geometric and dynamic constraints, has been an active area of research for many years. Very often the problem is divided into a high-level path planning stage where a feasible geometric path is determined and a low-level path following stage where system dynamics are taken into account. This paper combines both approaches for differentially flat systems into a single optimization problem. The geometric path is represented as a convex combination of two or more feasible paths and the dynamics of the system can subsequently be projected onto the path which leads to a single input system. The resulting optimization problem is transformed into a fixed end-time optimal control problem that can be initialized easily. Throughout the paper, the quadrotor, a challenging non-linear system, is used to illustrate the proposed approach.**

I. INTRODUCTION

The time-optimal path planning problem considers bringing a system from an initial state to a terminal state, while obeying the dynamic and geometric constraints. Solving this problem efficiently can be of great interest for maximizing productivity of robot systems, minimizing lap time in car racing and many other applications.

Due to the complexity of the problem, a decoupled approach is often followed, which solves the motion planning problem in two stages [1]. First a high-level planner determines a geometric path, while accounting for obstacle avoidance and other geometric constraints but ignoring low-level, dynamic constraints. Subsequently, in the path following stage, a time-optimal trajectory along the geometric path is determined taking into account dynamic constraints [2], [1], [3]. Although this approach requires lower computational effort, it often results in sub-optimal solutions.

In this paper, we consider the motion planning for differentially flat systems. For such systems, there exists a particular set of outputs that characterize all states and inputs of the system. Therefore, solving the motion planning problem through flatness avoids the integration of the (often highly nonlinear) differential equations. For differentially flat systems, an efficient formulation for the path following problem was recently derived in [4], [5]. Inspired by the problem formulation and the aforementioned two-level approach, a combined path planning–path following approach is proposed in this paper. The geometric path is parametrized as a convex combination of two or more outer paths, which are a function of a scalar path coordinate s with an unknown time dependency. Using a transformation of variables, the optimal control problem is formulated as a fixed end-time problem with s as independent variable that can be initialized easily.

Although the proposed method can be applied to any differentially flat system, the time-optimal flight of a quadrotor is the focus of this paper. Other approaches for the time-optimal control of a quadrotor have been proposed in [6], [7], [8]. The authors of [6] describe a direct method using discretized control inputs and solve the resulting optimization problem using genetic algorithms and nonlinear programming. In [7] a direct method is proposed that exploits the differential flatness of the system. Similar to our approach, the trajectory is parametrized as function of a scalar path coordinate with unknown time dependency. However, contrary to what the authors claim, the resulting optimization problem can be very challenging to solve and would require good initialization to ensure convergence. The authors of [8] use Pontryagin's minimum principle on a simplified two-dimensional quadrotor model to find time optimal trajectories. The approach, however, is limited to a simplified model and constraints to keep the computations tractable. Also, geometric constraints, as obstacles, are not considered. However, the solutions are analytic and provide an interesting benchmark for direct methods.

Section II describes first the concept of differential flatness followed by a general statement of the optimization problem in section II-B. Then, an efficient formulation of the path following problem is derived. Finally, section II-D describes our combined path planning–path following approach. Section III illustrates the developed methodology with several flight scenarios for a quadrotor and benchmarks them against [8].

In the following, we will use $\partial_\tau^k g(\tau)$ to denote the k-th derivative of $g(\tau)$ with respect to τ. For small k, we will use the shorthand notation $\dot{g}$ to denote the derivative with respect to time.

II. PROBLEM FORMULATION

A. Differential flatness

Before addressing the path following problem, let us first recall the notion of differential flatness, introduced in [9].

Definition (Differentially flat system). The system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{1}$$

*Goele Pipeleers is a Postdoctoral Fellow of the Research Foundation – Flanders (FWO – Vlaanderen). This work benefits from K.U.Leuven – BOF PFV/10/002 Center-of-Excellence Optimization in Engineering (OPTEC), the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO), research projects FP7–EMBOCON (ICT–248940), G.0377.09 of the Research Foundation – Flanders (FWO – Vlaanderen), and K.U.Leuven's Concerted Research Action GOA/10/11.

W. Van Loock, G. Pipeleers and J. Swevers are with the Department of Mechanical Engineering, Division PMA, KU Leuven, Celestijnenlaan 300B, 3001 Heverlee, Belgium wannes.vanloock at mech.kuleuven.be

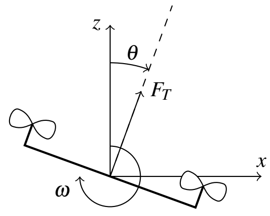

Fig. 1. Two-dimensional first principles quadrotor model and controls

with state $\mathbf{x} \in \mathbb{R}^n$ and input $\mathbf{u} \in \mathbb{R}^m$, $m \leq n$, is differentially flat if there exists a variable $\mathbf{z} \in \mathbb{R}^m$, the so-called flat output, such that

$$\mathbf{z}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \ldots, \partial_t^\alpha \mathbf{u}(t))$$

and

$$\begin{aligned} \mathbf{x}(t) &= \Phi(\mathbf{z}(t), \ldots, \partial_t^{\beta-1} \mathbf{z}(t)), \\ \mathbf{u}(t) &= \Psi(\mathbf{z}(t), \ldots, \partial_t^{\beta} \mathbf{z}(t)), \end{aligned} \tag{2}$$

where Φ and Ψ are smooth functions and $\alpha, \beta \in \mathbb{N}$. The flat output characterizes all the state motions and corresponding input history. It is a powerful concept in motion planning as it avoids integration of the differential equations. In the following, we consider systems of the form (1) with output $\mathbf{y} \in \mathbb{R}^m$. We assume that the system is differentially flat with flat output $\mathbf{z} = \mathbf{y}$.

Example (Two-dimensional first-principles quadrotor model [8]). The quadrotor is controlled by two inputs, the total thrust force F_T and the pitch rate ω, and has three degrees of freedom, the horizontal and vertical position x and z, and the pitch angle θ, as shown in Fig. 1.

The equations of motion are

$$\ddot{x} = \frac{F_T}{m} \sin\theta, \tag{3}$$

$$\ddot{z} = \frac{F_T}{m} \cos\theta - g, \tag{4}$$

$$\dot{\theta} = \omega, \tag{5}$$

where g denotes the gravitational acceleration and m is the quadrotor's mass. This system is differentially flat with the coordinate $\mathbf{y} = (x, z)$ as flat output and input $\mathbf{u} = (F_T, \omega)$. Indeed, from the first two equations it follows that the state

$$\theta = \arctan \frac{\ddot{x}}{\ddot{z} + g}.$$

We also find the system inputs as a function of the flat outputs

$$\begin{aligned} F_T &= m\sqrt{(\ddot{z}+g)^2 + \ddot{x}^2}, \\ \omega &= \frac{\dddot{x}(\ddot{z}+g) - \ddot{x}\dddot{z}}{(\ddot{z}+g)^2 + \ddot{x}^2}. \end{aligned}$$

Similarly, the full quadrotor model is differentially flat with flat outputs the coordinate of the center of gravity and the yaw angle [10].

B. Optimization problem

Consider a geometric path $\mathbf{y}_r(s) \in \mathscr{C}^\beta$, a regular, i.e. $\partial_s \mathbf{y}_r(s) \neq 0, \forall s \in [0,1]$, parametrized curve as a function of a scalar path coordinate s. The time dependency follows from the relation $s(t)$. Without loss of generality it is assumed that the trajectory starts at $t = 0$, ends at $t = T$ and that $s(0) = 0 \leq s(t) \leq s(T) = 1$. Furthermore, it is assumed that the velocity along the path is non-negative, i.e. $\dot{s}(t) \geq 0$ and that the boundary conditions $\mathbf{x}_0, \mathbf{x}_T$ are consistent with $\mathbf{y}_r(s)$, such that $\mathbf{x}_0 = \Phi(\mathbf{y}_r(0), \ldots, \mathbf{y}_r^{(\beta-1)}(0))$ and $\mathbf{x}_T = \Phi(\mathbf{y}_r(1), \ldots, \mathbf{y}_r^{(\beta-1)}(1))$.

In this paper we want to find an input signal $\mathbf{u}(t)$ and the a geometric path $\mathbf{y}_r(s)$ that allows for the fastest execution time such that (i) the geometric path is followed exactly by the system output $\mathbf{y}(t) = \mathbf{z}(t)$:

$$\mathbf{z}(t) = \mathbf{y}_r(s(t)),$$

(ii) without violating constraints on states and inputs

$$\mathbf{u}(t) \in \mathscr{U}, \, \mathbf{x}(t) \in \mathscr{X},$$

(iii) without violating geometric constraints

$$\mathbf{y}_r(s) \in \mathscr{Y},$$

and (iv) in minimal time T.

The problem can be seen as a combined path planning and path following approach. For differentially flat systems, the path following problem can be solved by projecting the system dynamics along the path onto a linear single-input system that is trivial to integrate [3], [4], [5], which avoids the integration of a high-dimensional non-linear state space model.

C. Optimal path following

Let us first fix the geometric reference $\mathbf{y}_r$ and formulate the time-optimal path following problem. In a first step, the system dynamics are projected along the path by applying the chainrule [4].

$$\begin{aligned} \dot{\mathbf{y}}_r(s) &= \partial_s \mathbf{y}_r(s)\dot{s}, \\ \ddot{\mathbf{y}}_r(s) &= \partial_s^2 \mathbf{y}_r(s)\dot{s}^2 + \partial_s \mathbf{y}_r(s)\ddot{s}, \end{aligned}$$

and so on, which allows us to rewrite states and input in terms of the path coordinate s and its time derivatives:

$$\begin{aligned} \mathbf{x} &= \Phi(s, \dot{s}, \ldots, \partial_t^{\beta-1} s), \\ \mathbf{u} &= \Psi(s, \dot{s}, \ldots, \partial_t^{\beta} s). \end{aligned}$$

Now, inspired by [3], instead of using the time t as independent variable, we rewrite states and inputs such that the path coordinate s is the independent variable. This is accomplished by the transformation of variables

$$\begin{aligned} \dot{s}^2 &= b(s), \\ \ddot{s} &= \frac{\partial_s b(s)}{2}, \\ \partial_t^3 s &= \frac{\partial_s^2 b(s)\dot{s}}{2} = \frac{\partial_s^2 b(s)\sqrt{b(s)}}{2}, \end{aligned} \tag{6}$$

and so on, such that [5]

$$\mathbf{x} = \Phi(\mathbf{y}_r(s), b(s), \ldots, \partial_s^{\beta-2} b(s)) = \Phi(s),$$
$$\mathbf{u} = \Psi(\mathbf{y}_r(s), b(s), \ldots, \partial_s^{\beta-1} b(s)) = \Psi(s).$$

Also, by using $dt = \frac{ds}{\dot{s}}$, the terminal time

$$T = \int_0^1 \frac{1}{\sqrt{b(s)}} ds. \tag{7}$$

With the above transformation the optimization problem is formulated as the optimal control problem with pseudo-time s, differential states $b(s)$ up to $\partial_s^{\beta-2} b(s)$ and control $\partial_s^{\beta-1} b(s)$

$$\begin{aligned}
\underset{b(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}} ds, \\
\text{subject to} \quad & \partial_s^i b(0) = b_{0_i},\, \partial_s^i s(1) = b_{T_i},\; i = 0, \ldots, \beta-2, \\
& b(s) \geq 0, \forall s \in [0,1], \\
& \Phi(s) \in \mathscr{X}, \forall s \in [0,1], \\
& \Psi(s) \in \mathscr{U}, \forall s \in [0,1],
\end{aligned} \tag{8}$$

where b_{0_i} and b_{T_i} are computed such that $\mathbf{x}_0 = \Phi(\mathbf{y}_r(0), b_{0_0}, \ldots, b_{0_{\beta-2}})$ and $\mathbf{x}_T = \Phi(\mathbf{y}_r(1), b_{T_0}, \ldots, b_{T_{\beta-2}})$. Note that we have obtained a fixed end-time optimal control problem, which, in general, is easier to solve than a free end-time problem. Indeed, the solution of free end-time problems solution can change strongly non-linearly with varying end-times.

In a second stage we also consider the geometric reference path as an unknown. How to deal with the geometric path and constraints is treated in the following section.

D. Combined optimal path planning and path following

Due to the possible complex geometry of the environment, imposing $\mathbf{y}_r(s) \in \mathscr{Y}$ directly in the optimization problem can pose quite a challenge. To circumvent this difficulty, the geometric reference path $\mathbf{y}_r(s)$ is defined as a convex combination of feasible paths $\mathbf{y}_i(s) \in \mathscr{Y}, i = 1, \ldots, k$:

$$\mathbf{y}_r(s) = \sum_{i=1}^{k} \mathbf{y}_i(s) p_i(s),$$

with

$$\sum_{i=1}^{k} p_i(s) = 1 \text{ and } p_i(s) \geq 0, \forall s \in [0,1],$$

and $p_i(s), \mathbf{y}_i(s) \in \mathscr{C}^\beta$. Fig. 2 shows two functions $\mathbf{y}_1(s)$ and $\mathbf{y}_2(s)$, which stay clear of the geometric constraints in the hatched areas. The geometric reference $\mathbf{y}_r(s)$ is defined as the convex combination of both functions, here with $p_1(s) = 1 - \sqrt{s}$ and $p_2(s) = \sqrt{s}$.

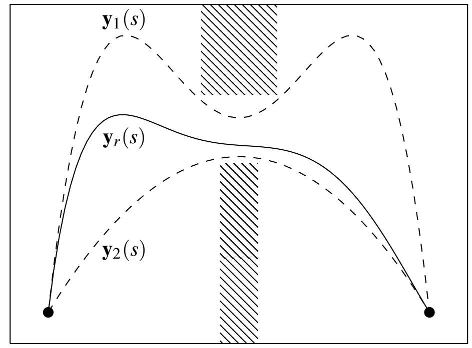

Fig. 2. Geometric reference $\mathbf{y}_r(s)$ as a convex combination of two paths $\mathbf{y}_1(s)$ and $\mathbf{y}_2(s)$ in a constrained environment

Similarly, the system dynamics can be projected along the path onto a linear single-input system:

$$\begin{aligned}
\dot{\mathbf{y}}_r(s) &= \sum_{i=1}^{k} \left(\partial_s \mathbf{y}_i(s) p_i(s) + \partial_s p_i(s) \mathbf{y}_i(s)\right) \sqrt{b(s)}, \\
\ddot{\mathbf{y}}_r(s) &= \sum_{i=1}^{k} \left(\partial_s \mathbf{y}_i(s) p_i(s) + \partial_s p_i(s) \mathbf{y}_i(s)\right) \partial_s b(s)/2 \\
&+ \sum_{i=1}^{k} \left(\partial_s^2 \mathbf{y}_i(s) p_i(s) + 2\partial_s p_i(s) \partial_s \mathbf{y}_i(s) + \partial_s^2 p_i(s) \mathbf{y}_i(s)\right) b(s),
\end{aligned} \tag{9}$$

and so on, such that we can rewrite the states and inputs to the system as

$$\mathbf{x} = \Phi(p_i, \partial_s p_i, \ldots, \partial_s^{\beta-1} p_i, b(s), \ldots, \partial_s^{\beta-2} b(s)) = \Phi(s),$$
$$\mathbf{u} = \Psi(p_i, \partial_s p_i, \ldots, \partial_s^{\beta} p_i, b(s), \ldots, \partial_s^{\beta-1} b(s)) = \Psi(s).$$

The path following problem (8) is extended with the control $\partial_s^\beta p_i(s)$, $i = 1, \ldots, k$ and states $p_i(s)$ up to $\partial_s^{\beta-1} p_i(s)$, $i = 1, \ldots, k$ to arrive at following optimal control problem

$$\begin{aligned}
\underset{b(\cdot), p_i(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}} ds, \\
\text{subject to} \quad & \partial_s^i b(0) = b_{0_i},\, \partial_s^i s(1) = b_{T_i},\; i = 0, \ldots, \beta-2, \\
& \sum_{i=1}^{k} p_i(s) = 1, \forall s \in [0,1], \\
& b(s) \geq 0, \forall s \in [0,1], \\
& p_i(s) \geq 0, \forall s \in [0,1], i = 1, \ldots, k, \\
& \Phi(s) \in \mathscr{X}, \forall s \in [0,1], \\
& \Psi(s) \in \mathscr{U}, \forall s \in [0,1].
\end{aligned} \tag{10}$$

Note that the optimal control problem does not require the integration of a possibly high-dimensional non-linear state space model and that it has been reduced to a smaller dimension. Indeed only the scalar functions $b(s)$ and $p_i(s)$ have to be determined. Moreover, the problem is transformed into a fixed end-time problem, which can be initialized easily.

E. Initialization

A typical issue when solving highly non-linear optimal control problems is the need for an accurate initial guess for the optimization algorithm to converge. In many cases it is non-trivial to even determine a feasible initial guess. Due to the geometric nature of our proposed formulation, however, a feasible initial guess is easily found. Consider the case of a rest-to-rest motion.

To impose zero state at start and end, we must make sure that $\partial_t \mathbf{y}_r^l(s)|_{0,T} = 0$, for $l = 1,\dots,\beta-1$. From equation (9), it follows that $\partial_s^l p_i(s)|_{0,1} = 0$ and $\partial_s^l \mathbf{y}_i(s)|_{0,1} = 0$, for $l = 1,\dots,\beta-1$ and $i = 1,\dots,k$ will impose a zero state on the boundaries. Note, that $\partial_s^l b(s) = 0$ for $l = 0,\dots,\beta-2$ cannot be imposed as this would render the goal function (7) infinity.

To initialize $p_i(s)$ we can now choose any positive function that adds up to one and obeys the constraints, for example

$$p_i(s) = \frac{1}{k},$$

the central path. For $b(s)$ we can choose any a small function $b(s) = \varepsilon(s)$ such that we travel quasi in steady state along the path, such that constraints are not violated. Initial guesses for other cases can be derived in similar fashion.

III. Benchmarks and Examples

In this section, we consider some simple quadrotor flight scenarios to illustrate the developed methodology and benchmark our method against [8]. Using the two-dimensional first principles quadrotor model (3), we demonstrate time-optimal horizontal and general displacements. The numerical model parameters are adopted from [8]. The thrust input is constrained to $1\,\mathrm{m\,s^{-2}} \le F_T/m \le 20\,\mathrm{m\,s^{-2}}$ and the rotational rate to $-10\,\mathrm{rad\,s^{-1}} \le \omega \le 10\,\mathrm{rad\,s^{-1}}$. In all examples, we start and stop in steady state, i.e.

$$\dot{x}(s) = \dot{z}(s) = \theta(s) = 0, \text{ for } s = 0,1.$$

$\theta(s) = 0$ implies that $\ddot{x}(s) = 0$ and $\ddot{z}+g > 0$. These constraints are added to the optimization problem either explicitly or implicitly by choosing the parameterizations of the outer paths, similar to [5].

The optimization problem is modeled using CasADi [11] and solved with Ipopt [12].

A. Horizontal displacement

Consider purely horizontal displacements from $(0,0)$ to $(x_T,0)$. The outer paths are defined as

$$\mathbf{y}_1 = (x_T q(s), -1) \text{ and } \mathbf{y}_2 = (x_T q(s), 1),$$

with $q(s) = s^5 + 5s^4(1-s) + 10s^3(1-s)^2$ a smooth polynomial such that $\partial_s^i q(s) = 0$ for $s = 0,1$ and $i = 1,2$. Also, the constraints $p_i(0) = p_i(1) = 0.5$ for $i = 1,2$ are added to the optimization problem such that the quadrotor starts and stops at $z = 0$.

Optimal times T are calculated for five distances $x_T = 3,6,9,12,15\,\mathrm{m}$ and compared to the travel times T^* reported in [8] in table I. The calculated times consistently outperform the results of [8] by a small amount. This is probably due to the explicit parametrization of the controls in [8], which are either at their maximum or minimum value or zero. The solver time for these trajectories is around 0.6 s for problem sizes around 1000 variables.

x_T [m]	T [s]	T^* [s]
3	0.8902	0.8977
6	1.223	1.231
9	1.478	1.488
12	1.694	1.705
15	1.885	1.895

TABLE I

Optimal times for horizontal displacements

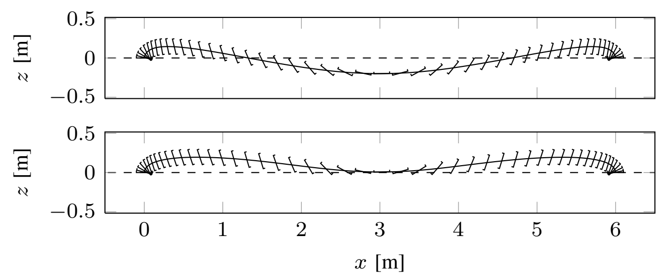

Fig. 3. Optimal horizontal maneuver with (bottom) and without (top) ground constraint. The quadrotors are drawn in equal time steps.

Assuming the quadrotor is taking of from the ground, these maneuvers would require the quadrotor to go underground as z becomes negative as is clear from Fig. 3 (top). Our method can easily cope with this constraint by defining the outer paths as

$$\mathbf{y}_1 = (x_T q(s), 0) \text{ and } \mathbf{y}_2 = (x_T q(s), 1),$$

ensuring a positive value for z. In [8], these geometric constraints cannot be taken into account. For $x_T = 6\,\mathrm{m}$ both trajectories are shown in Fig. 3. The trajectory time for the positive trajectory is 1.227 s, which is only marginally more compared to the unconstrained case.

B. General displacement

We consider general two-dimensional displacements from $(0,0)$ to (x_T, z_T). For geometrically unconstrained problems we define the outer paths as

$$\mathbf{y}_1 = (x_T q(s), -c) \text{ and } \mathbf{y}_2 = (x_T q(s), z_T + c),$$

where c is a chosen constant and constraints are added on $p_i(s)$ for $s = 0,1$ such that the trajectory start at $(0,0)$ and stops at (x_T, z_T).

As a benchmark, an optimal trajectory is calculated for $x_T = z_T = 5\,\mathrm{m}$. The constant c is chosen $c = 0\,\mathrm{m}$. The optimal trajectory is shown in Fig. 4. Our computations lead to a final time of 1.289 s, which is shorter than the 1.4 s reported in [8].

Moreover, in the proposed framework geometric constraints can be easily added to the optimization problem. Consider flying the quadrotor to $x_T = z_T = 5\,\mathrm{m}$ through a hoop with diameter 0.5 m, positioned vertically with its center at $(2.5\,\mathrm{m}, 2.5\,\mathrm{m})$. To ensure the quadrotor flies through the hoop, we define the outer paths as

$$\mathbf{y}_1 = \left(x_T q(s), (x_T r(s) - 2.5)^2 + 2.75\right)$$

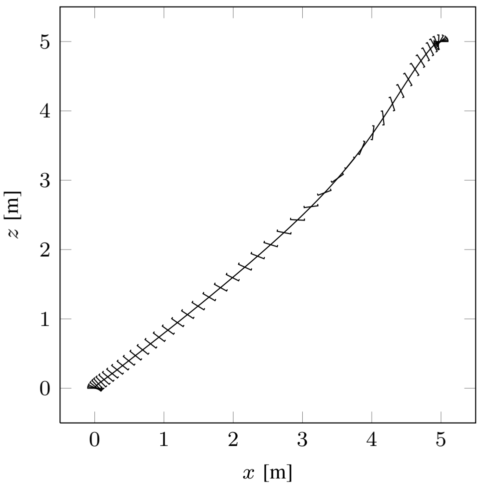

Fig. 4. Optimal general maneuver to $(5\,\mathrm{m}, 5\,\mathrm{m})$. The quadrotors are drawn in equal time steps.

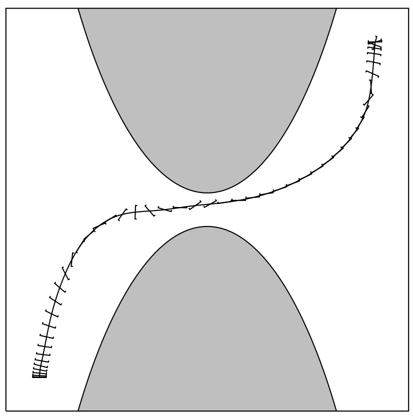

Fig. 5. Optimal constrained maneuver to $(5\,\mathrm{m}, 5\,\mathrm{m})$. The quadrotors are drawn in equal time steps.

and

$$\mathbf{y}_2 = \left(x_T q(s), -(x_T r(s) - 2.5)^2 - 2.25\right),$$

with $r(s) = s^3 + 3s^2(1-s)$ a smooth polynomial such that $\partial_s r(s) = 0$ for $s = 0, 1$. The optimal trajectory is shown in Fig. 5 along with the outer paths. The corresponding controls are shown in Fig. 6. Note the aggressive controls causing the quadrotor to flip and use its thrust for breaking.

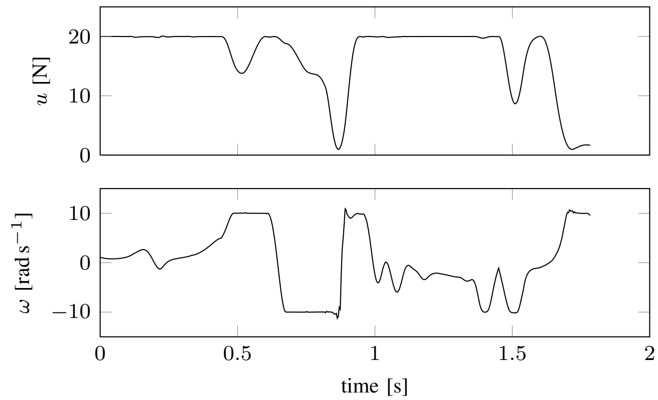

Fig. 6. Optimal controls for the constrained maneuver to $(5\,\mathrm{m}, 5\,\mathrm{m})$.

CONCLUSIONS

The paper proposes a framework to solve the time-optimal path planning problem for differentially flat systems. Hereto we developed a combined path planning, path following approach. The geometric path is represented as a convex combination of two or more feasible outer paths, which avoids complex geometric constraints in the optimization problem. Subsequently, the system dynamics are projected onto the parametrized path, which leads to a single input system. The resulting optimization problem is transformed into a fixed end-time optimal control problem which can be solved efficiently by interior-point solvers, such as Ipopt [12]. Several examples on a two-dimensional non-linear quadrotor model demonstrate the effectiveness of the proposed approach. Theoretical results from literature [8] are outperformed. Moreover, geometric constraints are easily added to the optimization problem, which allows modeling complex flight scenarios.

REFERENCES

[1] K. Shin and N. Mckay, "Minimum-time control of robotic manipulators with geometric path constraints," *Automatic Control, IEEE Transactions on*, vol. 30, no. 6, pp. 531–541, 1985.

[2] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[3] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *Automatic Control, IEEE Transactions on*, vol. 54, no. 10, pp. 2318 –2327, oct. 2009.

[4] T. Faulwasser, V. Hagenmeyer, and R. Findeisen, "Optimal exact path-following for constrained differentially flat systems," in *Proc. of 18th IFAC World Congress, Milano, Italy*, 2011, pp. 9875–9880.

[5] W. Van Loock, G. Pipeleers, M. Diehl, and J. Swevers, "Optimal path following for differentially flat systems," *Automatica*, submitted.

[6] L.-C. Lai, C.-C. Yang, and C.-J. Wu, "Time-optimal control of a hovering quad-rotor helicopter," *J. Intell. Robotics Syst.*, vol. 45, no. 2, pp. 115–135, Feb. 2006.

[7] Y. Bouktir, M. Haddad, and T. Chettibi, "Trajectory planning for a quadrotor helicopter," in *Control and Automation, 2008 16th Mediterranean Conference on*, june 2008, pp. 1258 –1263.

[8] M. Hehn, R. Ritz, and R. DAndrea, "Performance benchmarking of quadrotor systems using time-optimal control," *Autonomous Robots*, vol. 33, pp. 69–88, 2012.

[9] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, pp. 1327–1361, 1995.

[10] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[11] J. Andersson, J. Åkesson, and M. Diehl, "CasADi – A symbolic package for automatic differentiation and optimal control," in *Proc. 6th International Conference on Automatic Differentiation*, ser. Lecture Notes in Computational Science and Engineering, S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, Eds. Berlin: Springer, 2012.

[12] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006, 10.1007/s10107-004-0559-y.