This is the post peer-review accepted manuscript of:

S. Spedicato and G. Notarstefano, "Computing minimum-time trajectories for quadrotors via transverse coordinates," 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, 2016, pp. 239-244.

The published version is available online at:

https://doi.org/10.1109/CDC.2016.7798276

Computing minimum-time trajectories for quadrotors
via transverse coordinates

Sara Spedicato and Giuseppe Notarstefano*

Abstract— In this paper we present a novel strategy to compute minimum-time trajectories for quadrotors. In particular, we consider the motion in constrained environments, taking into account the physical limitations of the vehicle. Instead of approaching the optimization problem in its standard time-parameterized formulation, the proposed strategy is based on an appealing re-formulation. Transverse coordinates, expressing the distance from a "reference" path, are used to parameterize the vehicle position and a spatial parameter is used as independent variable. This re-formulation allows us to (i) obtain a fixed horizon problem and (ii) easily formulate (even complex) position constraints. The effectiveness of the proposed strategy is proven by numerical computations on two different illustrative scenarios.

I. INTRODUCTION

Numerous applications involving quadrotors require them to move inside areas characterized by physical boundaries, obstacles and even tight space constraints (as e.g., urban environments) in order to accomplish their robotics tasks. Trajectory generation, a core step for physical task realization [1], becomes extremely challenging in this scenario. A physically realizable trajectory must satisfy (i) the (nonlinear) system dynamics, (ii) the physical limits of the vehicle, and (iii) the position constraints. Moreover, the computation of optimal (rather than just feasible) trajectories is more and more becoming a necessity to ensure an efficient accomplishment of the tasks, posing an additional challenge in the trajectory generation problem.

Quadrotor trajectory planning in constrained environments can be addressed using a reacting (e.g., [2], [3]) or a planning approach. The majority of the planning algorithms, such as [4], [5], [6], take advantage of the differential flatness property to avoid the integration of nonlinear differential equations, reduce the order of the problem and simplify the definition of position constraints. The problem is posed in the flat output space, where outputs are approximated as a combination of primitives, such as polynomial functions [4], B-splines [5], or convex functions [6]. The parameters of the primitives are the optimization variables. When dealing with obstacle dense environments, trajectory generation is often performed using a decoupled approach ([7], [8], [9]) aiming to avoid the difficult definition of spatial constraints and the high computational cost. In a first stage, a collision-free path is generated by sampling-based path planning algorithms, such as the Rapidly-exploring Random Tree (RRT) in [7], [8] or the Probabilistic Roadmap (PRM) in [9], and without the dynamics constraint. In a second stage, an optimal trajectory (satisfying the system dynamics) is generated from the collision free path. Optimization techniques such as [10], [4], [5] can be used at this stage. Differently from these works, the one in [6] considers a space-parameterized problem reformulation, suitable for modeling complex flight scenarios, and thus avoids the decoupled approach. Finally, only few works do not consider the differential flatness hypothesis, such as [11]. Nevertheless, in [11] a discretized simple point-mass dynamics and approximated convex constraints are considered.

Our main contribution is the design of an optimization framework to generate feasible minimum-time quadrotor trajectories in geometrically constrained environments. In our problem setup (i) the vehicle nonlinear dynamics is directly included into the optimization problem (the differential flatness assumption is not required) and (ii) the vehicle position, which is an optimization variable, is not approximated via primitives. Similarly to [6], the minimum-time problem is formulated by using the path parameter as independent variable: a fixed horizon problem is obtained and even complex position constraints can be easily formulated. The optimization problem is solved combining the Projection Operator Newton method for Trajectory Optimization (PRONTO) [12] with a barrier function approach [13]. Similar problem setups and optimization methods are presented in [14] for two-wheeled mobile robots and in [15] for UAVs. In [14] a minimum-energy problem in a bi-dimensional space with obstacles is addressed by considering a standard formulation (i.e., with variables parameterized by the time). In [15] a virtual target approach (without position constraints) is presented using an arc-length parameterized transverse dynamics model. Differently from these works, we solve the optimization problem in the three-dimensional space and we define obstacle constraints by an arc-length parameterization. As an additional contribution, we present numerical computations on two challenging scenarios with respectively, a circular tube of variable section and a narrow corridor with an obstacle, as constrained spaces.

The paper is organized as follows. In Section II we present the standard formulation of the optimization problem we aim to solve. In Section III our trajectory generation strategy, based on an appealing reformulation of the problem, is described. Finally, in Section IV, we provide numerical computations discussing the minimum-time vehicle behavior.

* S. Spedicato and G. Notarstefano are with the Department of Engineering, Università del Salento, Via per Monteroni, 73100 Lecce, Italy, e-mail: name.lastname@unisalento.it.

II. The quadrotor minimum-time problem

We first briefly introduce the quadrotor model used in the paper and then recall the standard problem formulation.

The quadrotor dynamics can be described by the so called vectored-thrust dynamical model, [16],

$$\dot{\boldsymbol{p}} = \mathbf{v} \tag{1}$$

$$\dot{\mathbf{v}} = g\boldsymbol{e}_3 - fm^{-1}R(\boldsymbol{\Phi})\boldsymbol{e}_3 \tag{2}$$

$$\dot{\boldsymbol{\Phi}} = J\boldsymbol{\omega} \tag{3}$$

$$\dot{\boldsymbol{\omega}} = -I^{-1}\Omega I\boldsymbol{\omega} + I^{-1}\boldsymbol{\gamma}, \tag{4}$$

where $\boldsymbol{p} = [p_1 \; p_2 \; p_3]^T$ is the vehicle position expressed in the inertial frame $\mathcal{F}_i$, $\mathbf{v} = [\mathrm{v}_1 \; \mathrm{v}_2 \; \mathrm{v}_3]^T$ is the linear velocity expressed in $\mathcal{F}_i$, $\boldsymbol{\omega} = [\omega_1 \; \omega_2 \; \omega_3]^T$ is the angular rate expressed in the body frame $\mathcal{F}_b$ ($\Omega \in so(3)$ is such that, for $\boldsymbol{\beta} \in \mathbb{R}^3, \Omega\boldsymbol{\beta} = \boldsymbol{\omega} \times \boldsymbol{\beta}$), and $J \in \mathbb{R}^{3\times 3}$ is the Jacobian matrix. The rotation matrix $R(\boldsymbol{\Phi}) \in SO(3)$, mapping vectors in $\mathcal{F}_b$ into vectors in $\mathcal{F}_i$, is parameterized using roll-pitch-yaw angles, denoted by $\boldsymbol{\Phi} := [\varphi \; \theta \; \psi]^T$. Furthermore, m and I are respectively the mass and the inertia matrix, g is the gravity constant, and $\boldsymbol{e}_3 = [0 \; 0 \; 1]^T$. The vehicle is controlled by the thrust f and the torques $\boldsymbol{\gamma}$.

For the vehicle maneuvering, we use a cascade control scheme with an off-board position/attitude control loop and an on-board angular rate controller. Assuming that the *virtual* control input $\boldsymbol{\omega}$ is tracked by the on-board angular rate controller, we restrict our trajectory generation problem on the position/attitude subsystem (1-3), which can be written in state-space form as

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{5}$$

with state $\boldsymbol{x} = [\boldsymbol{p}^T \; \mathbf{v}^T \; \boldsymbol{\Phi}^T]^T$ and input $\boldsymbol{u} = [\boldsymbol{\omega}^T \; f]^T$.

We deal with the following optimal control problem:

$$
\begin{aligned}
\min_{\boldsymbol{x}(\cdot), \boldsymbol{u}(\cdot)} & \int_0^T 1 \, d\tau \\
\text{subj. to } & \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \boldsymbol{x}(0) = \boldsymbol{x_0} \quad \textit{(dynamics)} \\
& |\omega_i(t)| \le \omega_{i,max}, \; i = 1, 2, 3, \textit{ (angular rate)} \\
& 0 < f_{min} \le f(t) \le f_{max} \textit{ (thrust)} \\
& |\varphi(t)| \le \varphi_{max}, \; |\theta(t)| \le \theta_{max} \textit{ (angles)} \\
& c_{obs}(\boldsymbol{p}(t)) \le 0 \textit{ (position constraints)}
\end{aligned} \tag{6}
$$

where $\omega_{i,max}$ are the bounds on the angular rate, f_{min} and f_{max} are lower an upper bounds on thrust, φ_{max} and θ_{max} are the bounds on the roll and pitch angles respectively, and $c_{obs} : \mathbb{R}^3 \to \mathbb{R}$ represents position constraints taking into account physical boundaries and obstacles. The position constraints may also represent GPS denied areas or spaces with limited communication. Angular rates have bounds, due to, for example, a limited range of the gyroscopes or to controller limitations. The vehicle thrust is also limited: quadrotor vehicles can only generate positive thrust and the maximum rotor speed is limited. Furthermore, constraints on angles are imposed into the optimization problem in order to avoid acrobatic vehicle configurations.

III. Minimum-Time Trajectory Generation Strategy

In order to solve the minimum-time problem, we consider an equivalent formulation of (6) where a spatial parameter (instead of time) is the independent variable. The derivation of the equivalent problem formulation is based on three steps: (i) generation of a "reference" path, (ii) definition of the transverse dynamics, in which the vehicle position is parameterized by transverse coordinates (with respect to that path), and (iii) mapping of time-parameterized cost and constraints into arc-length parameterized ones. The PRONTO method, combined with a barrier function approach, is then used to numerically solve the optimization problem.

A. Minimum-Time problem: equivalent formulation

The first step to convert the standard problem into the new formulation is the computation of a "reference" path $\bar{\boldsymbol{p}}_r(s)$, $\forall s \in [0, L]$, where s is the arc-length of the path and L is its total length. In the following, we denote the arc-length parameterized functions with a bar, and the derivative with respect to the arc-length with a prime, i.e., $\bar{\boldsymbol{p}}_r'(s) := d\bar{\boldsymbol{p}}_r(s)/ds$. The "reference" path is computed as a C^∞ geometric curve, e.g., using arctangent functions, as in our numerical computations. Note that $\bar{\boldsymbol{p}}_r(\cdot)$ is not required to satisfy the position constraints, and it can be suitably chosen.

With $\bar{\boldsymbol{p}}_r(\cdot)$ in hand, we re-write the system dynamics (5) as follows.

First, we design a change of coordinates from the inertial position $\boldsymbol{p} \in \mathbb{R}^3$ to the transverse coordinate vector $\boldsymbol{w} \in \mathbb{R}^2$, such that $\boldsymbol{w} = [w_1 \; w_2]^T$. Let us consider the "reference" path $\bar{\boldsymbol{p}}_r(\cdot)$. A Serret-Frenet frame, whose origin has $\bar{\boldsymbol{p}}_r(s)$ as coordinates, can be defined $\forall s \in [0, L]$. In particular, the tangent, normal and bi-normal vectors, respectively $\bar{\boldsymbol{t}}(s), \bar{\boldsymbol{n}}(s), \bar{\boldsymbol{b}}(s)$, are defined, with components in the inertial frame, as

$$\bar{\boldsymbol{t}}(s) := \bar{\boldsymbol{p}}_r'(s), \tag{7}$$

$$\bar{\boldsymbol{n}}(s) := \frac{\bar{\boldsymbol{p}}_r''(s)}{\bar{k}(s)}, \tag{8}$$

$$\bar{\boldsymbol{b}}(s) := \bar{\boldsymbol{t}}(s) \times \bar{\boldsymbol{n}}(s), \tag{9}$$

where $\bar{k}(s) := \|\bar{\boldsymbol{p}}_r''(s)\|_2$ is the curvature of $\bar{\boldsymbol{p}}_r(\cdot)$ at s. Moreover, we define the rotation matrix $\bar{R}_{SF} := [\bar{\boldsymbol{t}} \; \bar{\boldsymbol{n}} \; \bar{\boldsymbol{b}}]$ mapping vectors with components in the Serret-Frenet frame into vectors with components in the inertial frame. According to the Serret-Frenet formulas, the arc-length derivative of the Serret-Frenet rotation matrix is

$$\bar{R}_{SF}'(s) = \bar{R}_{SF}(s) \begin{bmatrix} 0 & -\bar{k}(s) & 0 \\ \bar{k}(s) & 0 & -\bar{\tau}(s) \\ 0 & \bar{\tau}(s) & 0 \end{bmatrix}, \tag{10}$$

where $\bar{\tau}(s) := \bar{\boldsymbol{n}}(s) \, \bar{\boldsymbol{b}}'(s)$ is the torsion of $\bar{\boldsymbol{p}}_r(\cdot)$ at s. Let us now consider the quadrotor center of mass with position $\boldsymbol{p}$. As depicted in Figure 1, its orthogonal projection on the "reference" path identifies a point with position $\bar{\boldsymbol{p}}_r(\pi(\boldsymbol{p}))$, where the function $\pi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is defined as

$$\pi(\boldsymbol{p}) := \arg\min_{s \in \mathbb{R}_0^+} \|\boldsymbol{p} - \bar{\boldsymbol{p}}_r(s)\|^2. \tag{11}$$

Note that, the minimizing arc-length is unique provided that $\bar{\boldsymbol{p}}_r(\cdot)$ is locally a non-intersecting C^2 curve with non-vanishing $\bar{\boldsymbol{p}}'_r(\cdot)$. By mapping $\boldsymbol{p} - \bar{\boldsymbol{p}}_r(\pi(\boldsymbol{p}))$ into a vector with components in the Serret-Frenet frame attached to $\bar{\boldsymbol{p}}_r(\pi(\boldsymbol{p}))$, we obtain $\boldsymbol{d}_{SF} := R_{SF}(\pi(\boldsymbol{p}))^T(\boldsymbol{p} - \bar{\boldsymbol{p}}_r(\pi(\boldsymbol{p})))$. Moreover, noticing that the component related to the tangent vector is always zero by construction, we define the transverse coordinates as the second and third components of $\boldsymbol{d}_{SF}$. Thus $\boldsymbol{d}_{SF} := [0 \ w_1 \ w_2]^T$. More formally, we define the locally invertible function $\phi : \mathbb{R}^3 \to \mathbb{R}^2$, such that $\boldsymbol{w} = \phi(\boldsymbol{p})$, with components defined as

$$\phi_1(\boldsymbol{p}) := \bar{\boldsymbol{n}}(\pi(\boldsymbol{p}))^T(\boldsymbol{p} - \bar{\boldsymbol{p}}_r(\pi(\boldsymbol{p}))), \qquad (12)$$

$$\phi_2(\boldsymbol{p}) := \bar{\boldsymbol{b}}(\pi(\boldsymbol{p}))^T(\boldsymbol{p} - \bar{\boldsymbol{p}}_r(\pi(\boldsymbol{p}))). \qquad (13)$$

Moreover, we denote by $\mathbf{v}_{SF} := \bar{R}_{SF}(\pi(\boldsymbol{p}))^T \mathbf{v}$ the linear velocity of the quadrotor with components in the Serret-Frenet frame. In particular, $\mathbf{v}_{SF} = [\mathbf{v}_t \ \mathbf{v}_n \ \mathbf{v}_b]^T$.
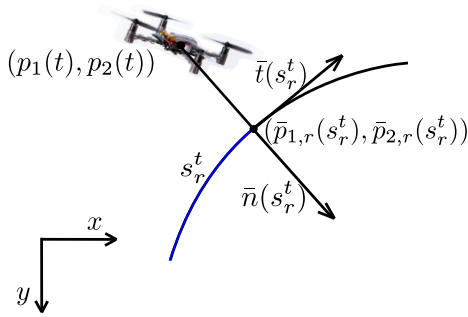

Fig. 1: Selection of the arc-length s_r^t identifying the point on the "reference" path at minimum distance from the quadrotor position at the time instant t.

Second, we note that, considering the position function $\boldsymbol{p}(\cdot)$ with the time t as independent variable, the function $s_r := \pi \circ \boldsymbol{p}$ provides a change of variables from the time t to the arc-length s. In particular, we denote by $s_r^t := s_r(t)$ the function $s_r(\cdot)$ evaluated at the time t, and by $\dot{s}_r^t := \dot{s}_r(t)$ the time derivative of $s_r(\cdot)$, evaluated at t. A generic arc-length function $\bar{\alpha}(\cdot)$ can thus be expressed as the time function $\bar{\alpha}(s_r(\cdot))$. Note that, the time derivative of $\bar{\alpha}(s_r(\cdot))$ is $\frac{d\bar{\alpha}(s_r(t))}{dt} = \bar{\alpha}'(s_r^t) \dot{s}_r^t$. Using the function $s_r : \mathbb{R}_0^+ \mapsto \mathbb{R}_0^+$, the position of the quadrotor center of mass $\boldsymbol{p}(t)$, at time instant t, can be written as

$$\boldsymbol{p}(t) = \bar{\boldsymbol{p}}_r(s_r^t) + \bar{R}_{SF}(s_r^t) \, \boldsymbol{d}_{SF}(t). \qquad (14)$$

Differentiating (14) with respect to time, we get

$$\mathbf{v}(t) = \bar{\boldsymbol{p}}'_r(s_r^t) \, \dot{s}_r^t + \bar{R}'_{SF}(s_r^t) \, \dot{s}_r^t \, \boldsymbol{d}_{SF}(t) + \bar{R}_{SF}(s_r^t) \, \dot{\boldsymbol{d}}_{SF}(t). \qquad (15)$$

Multiplying both members of equation (15) by $\bar{R}_{SF}^T$, using (10), the definition of $\mathbf{v}_{SF}$, and $\bar{\boldsymbol{p}}'_r(s_r^t) = \bar{R}_{SF}(s_r^t) [1 \ 0 \ 0]^T$, we get

$$\dot{s}_r^t = \frac{\mathbf{v}_t(t)}{1 - \bar{k}(s_r^t)w_1(t)} \qquad (16)$$

$$\dot{w}_1(t) = \mathbf{v}_n(t) + \bar{\tau}(s_r^t)\dot{s}_r^t w_2(t) \qquad (17)$$

$$\dot{w}_2(t) = \mathbf{v}_b(t) - \bar{\tau}(s_r^t)\dot{s}_r^t w_1(t). \qquad (18)$$

Furthermore, since we are using transverse coordinates to parameterize the vehicle position, we re-write equation (2) using $\mathbf{v}_{SF}$ instead of $\mathbf{v}$. Differentiating both members of $\mathbf{v}(t) = \bar{R}_{SF}(s_r^t)\mathbf{v}_{SF}(t)$ with respect to time, and using (2) and (10), we get

$$\dot{\mathbf{v}}_{SF} = \bar{R}_{SF}^T \dot{\mathbf{v}} - \begin{bmatrix} 0 & -\bar{k} & 0 \\ \bar{k} & 0 & -\bar{\tau} \\ 0 & \bar{\tau} & 0 \end{bmatrix} \dot{s}_r \mathbf{v}_{SF}, \qquad (19)$$

with $\dot{\mathbf{v}} = (g\boldsymbol{e}_3 - fm^{-1}R(\boldsymbol{\Phi})\boldsymbol{e}_3)$. The dependence on t and s_r^t is omitted for simplicity. Equations (17), (18), (19) and (3) can be compactly written as

$$\dot{\boldsymbol{q}}_S(t) = \tilde{f}(\boldsymbol{q}_S(t), \boldsymbol{u}(t)), \qquad (20)$$

with state $\boldsymbol{q}_S = [\boldsymbol{w}^T \ \mathbf{v}_{SF}^T \ \boldsymbol{\Phi}^T]^T$ and input $\boldsymbol{u} = [\boldsymbol{\omega}^T \ f]^T$.

Third and final, we write the dynamics (20) in terms of the arc-length s. By using the chain rule, we have $\dot{\boldsymbol{q}}_S(t) = \bar{\boldsymbol{q}}'_S(s_r^t)\dot{s}_r^t$. Moreover, we parameterize the state $\boldsymbol{q}_S$ and the input $\boldsymbol{u}$ using s_r^t, i.e., $\boldsymbol{q}_S(t) = \bar{\boldsymbol{q}}_S(s_r^t)$ and $\boldsymbol{u}(t) = \bar{\boldsymbol{u}}(s_r^t)$. Thus, equations (20) can be written as the *transverse dynamics*

$$\bar{\boldsymbol{q}}'_S(s) = \bar{f}(\bar{\boldsymbol{q}}_S(s), \bar{\boldsymbol{u}}(s)),$$

with the arc-length s as independent variable and where

$$\bar{f}(\bar{\boldsymbol{q}}_S(s), \bar{\boldsymbol{u}}(s)) := \tilde{f}(\bar{\boldsymbol{q}}_S(s), \bar{\boldsymbol{u}}(s)) \frac{1 - \bar{k}(s)\bar{w}_1(s)}{\bar{\mathbf{v}}_t(s)}.$$

Re-writing the time-parameterized cost and constraints of (6) into arc-length parameterized ones, the new formulation of the minimum-time problem, using the new set of coordinates $(\bar{\boldsymbol{q}}_S, \bar{\boldsymbol{u}})$ and with s as independent variable, is

$$\min_{\bar{\boldsymbol{q}}_S(\cdot), \bar{\boldsymbol{u}}(\cdot)} \int_0^L \frac{1 - \bar{k}(s)\bar{w}_1(s)}{\bar{\mathbf{v}}_t(s)} \, ds$$

subj. to $\bar{\boldsymbol{q}}'_S = \bar{f}(\bar{\boldsymbol{q}}_S(s), \bar{\boldsymbol{u}}(s)), \quad \bar{\boldsymbol{q}}_S(0) = \boldsymbol{q}_{S0}$ *(dynamics)*

$\left(\dfrac{\bar{\omega}_i(s)}{\omega_{i,max}}\right)^2 - 1 \le 0, \ i = 1, 2, 3$ *(angular rate)*

$\left(\dfrac{2\bar{f}(s) - (f_{max} + f_{min})}{(f_{max} - f_{min})}\right)^2 - 1 \le 0$ *(thrust)*

$\left(\dfrac{\bar{\varphi}(s)}{\bar{\varphi}_{max}(s)}\right)^2 - 1 \le 0$ *(roll angle)*

$\left(\dfrac{\bar{\theta}(s)}{\bar{\theta}_{max}(s)}\right)^2 - 1 \le 0$ *(pitch angle)*

$\bar{c}_{obs}(\bar{\boldsymbol{w}}(s)) \le 0$ *(position constraints)*.

(21)

The constraints on the roll and pitch angles, the thrust and the angular rates are re-written by means of an s-parameterized equivalent form (with smooth functions). The position constraints $\bar{c}_{obs}(\bar{\boldsymbol{w}}(s)) \le 0$ have specific formulations according to the shape of the collision free region. Considering regions with rectangular or circular transversal sections, we get the following inequalities. For rectangular sections, the position constraints are

$$\left(\frac{2\bar{w}_i(s) - (\bar{w}_{i,max}(s) + \bar{w}_{i,min}(s))}{(\bar{w}_{i,max}(s) - \bar{w}_{i,min}(s))}\right)^2 - 1 \le 0, \qquad (22)$$

$\forall s \in [0,L]$, $\forall i = 1,2$, where the non constant functions $\bar{w}_{i,min}(\cdot)$ and $\bar{w}_{i,max}(\cdot)$ identify the obstacle boundaries. For circular sections, the position constraint is

$$\left(\frac{||\bar{\boldsymbol{w}}(s)||}{\bar{r}_{obs}(s)}\right)^2 - 1 \le 0, \quad \forall s \in [0,L], \tag{23}$$

where the non constant function $\bar{r}_{obs}(\cdot)$ identifies the circular boundary of the collision free region.

The new formulation (21) is equivalent to (6) since trajectories solving (21) can be mapped into trajectories solving (6). Using transverse coordinates, the formulation of position constraints is simpler than using the time-parameterized inertial frame position. Since collision free region boundaries $\bar{w}_{i,min}(\cdot)$, $\bar{w}_{i,max}(\cdot)$ and $\bar{r}_{obs}(\cdot)$ are arc-length functions, even complex spaces can be modelled.

B. Numerical solution to the optimization problem

In order to solve problem (21), we use a combination of the PRojection Operator based Newton method for Trajectory Optimization (PRONTO) [12] with a barrier function approach [13]. We relax state-input constraints by adding them in the cost functional, i.e., given the state-input curve $\xi = (\bar{\boldsymbol{q}}_S(\cdot), \bar{\boldsymbol{u}}(\cdot))$, we consider the problem

$$\min_{\xi} \; h(\xi; \bar{k}) + \epsilon b_\nu(\xi)$$
$$\text{subj. to } \bar{\boldsymbol{q}}'_S(s) = \bar{f}(\bar{\boldsymbol{q}}_S(s), \bar{\boldsymbol{u}}(s)), \quad \bar{\boldsymbol{q}}_S(0) = \boldsymbol{q}_{S0} \tag{24}$$

where

$$h(\xi; \bar{k}) = \int_0^L \frac{1 - \bar{k}(s)\bar{w}_1(s)}{\bar{v}_t(s)} \, ds$$

$$b_\nu(\xi) = \int_0^L \sum_j \beta_\nu(-c_j(\bar{\boldsymbol{q}}_S(s), \bar{\boldsymbol{u}}(s))) \, ds$$

$$\beta_\nu(x) = \begin{cases} -\log(x) & x > \nu \\ -\log(\nu) + \frac{1}{2}\left[(\frac{x-2\nu}{\nu})^2 - 1\right] & x \le \nu, \end{cases}$$

and $c_j(\bar{\boldsymbol{q}}_S(s), \bar{\boldsymbol{u}}(s))$, j=1,..,7 denote the constraints in problem (21). The strategy to find a solution to (21) consists in iteratively solving problem (24) by reducing the parameters ϵ and ν at each iteration, and thus pushing the trajectory towards the constraint boundaries. Each instance of problem (24) is solved by means of the PRONTO algorithm. This is based on a properly designed *projection operator* $\mathcal{P} : \xi_c \to \xi$, mapping a state-control curve $\xi_c = (\bar{\boldsymbol{q}}_c(\cdot), \bar{\boldsymbol{u}}_c(\cdot))$ into a system trajectory ξ, by the nonlinear feedback system

$$\bar{\boldsymbol{q}}'_S(s) = \bar{f}(\bar{\boldsymbol{q}}_S(s), \bar{\boldsymbol{u}}(s)), \quad \bar{\boldsymbol{q}}_S(0) = \boldsymbol{q}_{S0}$$
$$\bar{\boldsymbol{u}}(s) = \bar{\boldsymbol{u}}_c(s) + \bar{K}(s)(\bar{\boldsymbol{q}}_c(s) - \bar{\boldsymbol{q}}_S(s)), \tag{25}$$

where the feedback gain $\bar{K}(\cdot)$ is designed by solving a suitable linear quadratic optimal control problem. The projection operator is used to convert the dynamically constrained optimization problem (24) into the unconstrained problem

$$\min_{\xi} g(\xi; \bar{k}), \tag{26}$$

where $g(\xi; \bar{k}) = h(\mathcal{P}(\xi); \bar{k}) + \epsilon b_\nu(\mathcal{P}(\xi))$. Then, using an (infinite dimensional) Newton descent method, a local minimizer of (26) is computed.

An initial trajectory, with a position matching the "reference" path, is used to initialize the algorithm. Given the current trajectory iterate ξ_i, the search direction ζ_i is obtained by solving a linear quadratic optimal control problem with cost $Dg(\xi_i; \bar{k}) \cdot \zeta + \frac{1}{2}D^2 g(\xi_i; \bar{k})(\zeta, \zeta)$, where $\zeta \mapsto Dg(\xi_i; \bar{k}) \cdot \zeta$ and $\zeta \mapsto D^2 g(\xi_i; \bar{k})(\zeta, \zeta)$ are respectively the first and second Fréchet differentials of the functional $g(\xi, \bar{k})$ at ξ_i. Then, the curve $\xi_i + \gamma_i \zeta_i$, where γ_i is a step size obtained through a standard backtracking line search, is projected, by means of the projection operator, in order to get a new trajectory ξ_{i+1}. The strength of this approach is that the local minimizer of (26) is obtained as the limit of a sequence of trajectories, i.e., curves satisfying the dynamics. Furthermore, the feedback system (25), defining the projection operator, allows us to generate trajectories in a numerically stable manner.

IV. NUMERICAL COMPUTATIONS

In this section, we present numerical computations to show the effectiveness of the proposed strategy.

We consider two different scenarios with a nano-quadrotor having mass $m = 0.021$ kg.

In the first scenario, the vehicle moves inside a tube with a 3D center line and circular transversal sections getting smaller along it. Results are depicted in Figures 2 and 3. As regards the initial trajectory (dot-dashed green line), by choosing as "reference" path the 3D center-line of the tube, the collision free region is defined by constraint (23) where $\bar{r}_{obs}(\cdot)$ is depicted in Figure 2b (gray line). Furthermore, we set an initial constant velocity and a zero yaw angle. The remaining initial states and inputs are computed by tracking these positions and velocities. Note that, in real applications, the bounds of the collision free region should be restricted (with respect to the physical boundaries of the region) in order to consider the vehicle dimension and an imperfect tracking of the trajectory. Each intermediate optimal trajectory (dotted black line) is computed by iteratively solving problem (24) with decreasing values of ϵ and ν. As regards the minimum-time trajectory (blue line), the maneuver is performed in 0.36 s, the optimal path touches the tube at arc-lengths 4.11 m, 9.35 m and 11.88 m (Figure 2b) and the velocity $\bar{v}_t$ (Figure 2c) reaches more than 7 m/s at the end of the path. As regards inputs, while constraints on thrust (Figure 3d) are always active, roll and pitch rate (Figures 3a and 3b, respectively) are saturated after $s = 1$ m and the yaw rate (Figure 3c) touches the constraint boundary only in the beginning of the maneuver.

In the second scenario, the vehicle moves from one room to another through a narrow corridor with a squared section with side length 0.5 m. Leaving the corridor, the vehicle encounters an obstacle and is forced to fly above it, having a final altitude $\bar{p}_3(L) \le -1$ m. In this scenario, as an additional requirement, we seek for a trajectory reaching the desired final state $\bar{\boldsymbol{q}}_S^d = [0 \; -1.5 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0]^T$. In order to fulfil this objective, we use a penalty function approach by adding the term $\frac{1}{2}(\bar{\boldsymbol{q}}_S^d - \bar{\boldsymbol{q}}_S(L))^T \rho I \, (\bar{\boldsymbol{q}}_S^d - \bar{\boldsymbol{q}}_S(L))$, where ρ is a penalty parameter and I is the identity 8×8 matrix, in the cost functional of problem (21). Results are depicted in

(a) Path

(b) Distance from the "reference" path

(c) Velocity $\bar{v}_t$

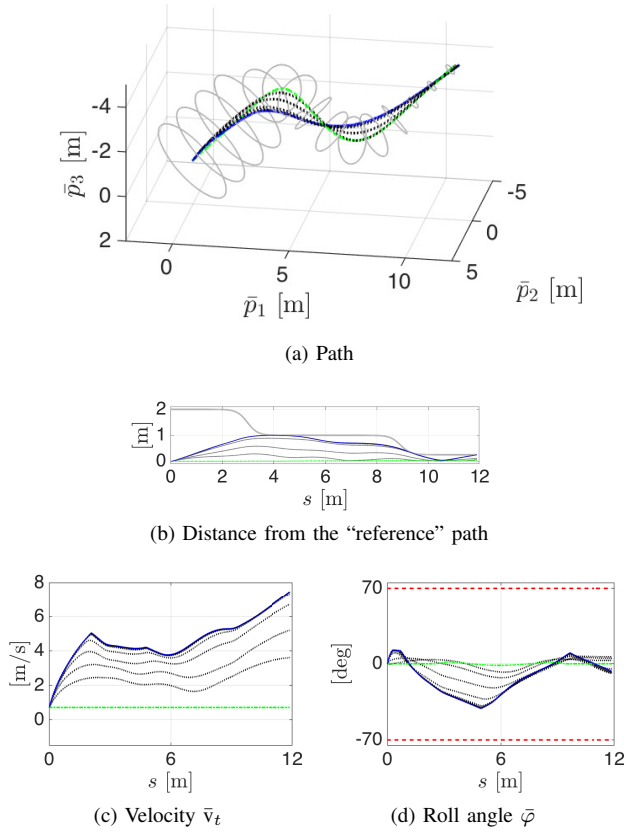(d) Roll angle $\bar{\varphi}$

Fig. 2: Scenario 1 (path, distance from the "reference" path and states). Initial (dot-dashed green), intermediate (dotted black) and minimum-time (solid blue) trajectory. Tube boundaries (gray) and angular constraints (dashed red).

(a) Roll rate $\bar{\omega}_1$

(b) Pitch rate $\bar{\omega}_2$

(c) Yaw rate $\bar{\omega}_3$
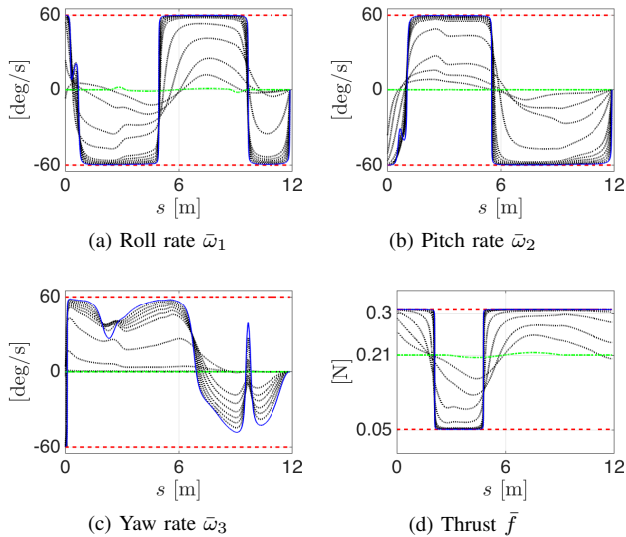

(d) Thrust $\bar{f}$

Fig. 3: Scenario 1 (inputs). Initial (dot-dashed green), intermediate (dotted black) and minimum-time (solid blue) trajectory. Constraint boundaries (dashed red).

Figures 4 and 5. We choose as "reference" path an unfeasible curve on the $\bar{p}_1 - \bar{p}_2$ plane and the collision free region is defined by constraint (22) where obstacle boundaries $\bar{w}_{i,min}(\cdot)$ and $\bar{w}_{i,max}(\cdot)$, $i = 1, 2$ are depicted in gray in Figures 4b and 4c. In particular, $\bar{w}_{2,min}$ is defined in order to take into account the presence of the obstacle that forces the vehicle to fly above it. An initial constant velocity and a zero yaw angle are considered and the remaining initial states and inputs are computed as in the first scenario. Note that, even if the initial trajectory is unfeasible, intermediate trajectories are feasible. As regards the minimum-time trajectory, the maneuver is performed in 0.5 s and the path touches the constraint boundaries when the vehicle is inside the corridor (Figures 4b and 4c). The velocity $\bar{v}_t$ (Figure 4d) reaches a peak of 9 m/s in the middle of the path and approaches zero at the end. As regards the inputs, only constraints on roll and pitch rates (Figures 5a, 5b, respectively) are always active. Furthermore, note that the desired final state is not exactly reached, since our penalty function approach only guarantees approximated solutions. An ad-hoc strategy should be designed to exactly reach the desired final state.

(a) Path: 3D view

(b) Transverse coordinate $\bar{w}_1$

(c) Transverse coordinate $\bar{w}_2$

(d) Velocity $\bar{v}_t$
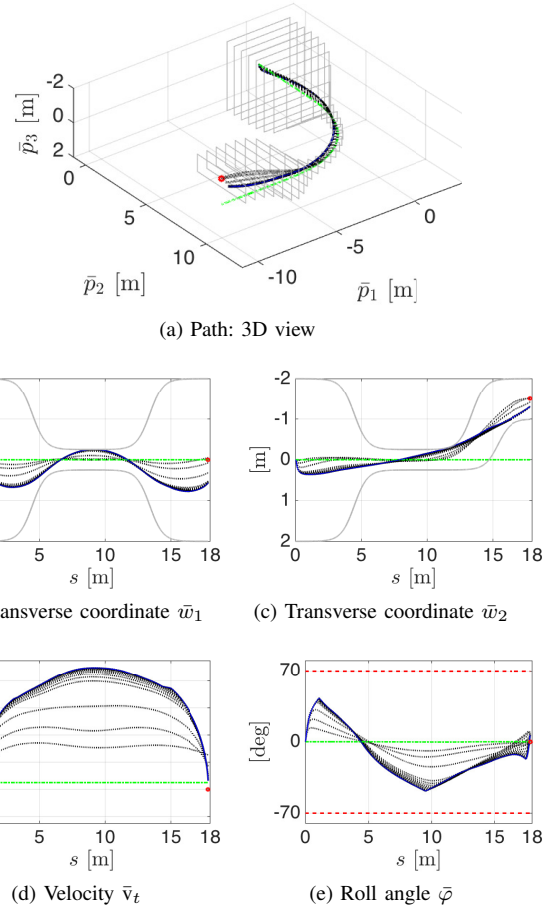

(e) Roll angle $\bar{\varphi}$

Fig. 4: Scenario 2 (path and states). Initial (dot-dashed green), intermediate (dotted black) and minimum-time (solid blue) trajectory. Collision free region boundary (gray), angular constraints (dashed red) and final desired state (red circle).

(a) Roll rate $\bar{\omega}_1$

(b) Pitch rate $\bar{\omega}_2$

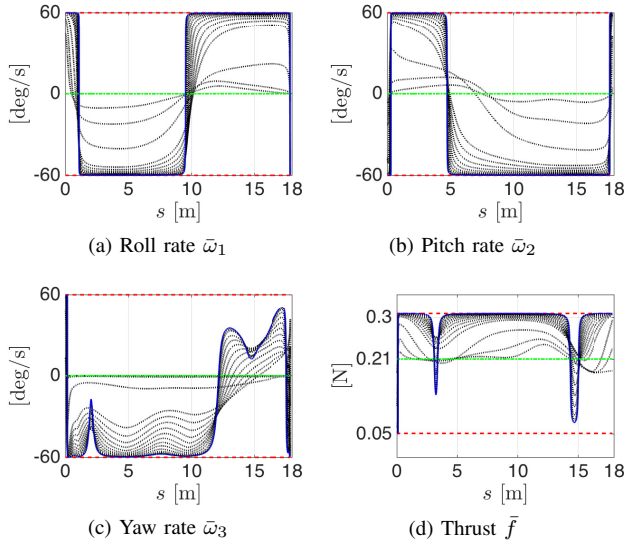(c) Yaw rate $\bar{\omega}_3$

(d) Thrust $\bar{f}$

Fig. 5: Scenario 2 (inputs). Initial (dot-dashed green), intermediate (dotted black) and minimum-time (solid blue) trajectory. Constraint boundaries (dashed red).

V. CONCLUSION

In this paper, we have presented a strategy to compute minimum-time trajectories for quadrotors in constrained spaces. Our approach consists of: (i) generating a "reference" path, (ii) expressing the vehicle dynamics using transverse coordinates, and (iii) re-defining constraints in the new coordinates. We obtain a re-formulation of the problem, which we solve by combining the PRONTO algorithm with a barrier function approach. Numerical computations on two challenging scenarios prove the effectiveness of the strategy. As a future work, we aim to compare our numerical results with a theoretical analysis on particular scenarios.

REFERENCES

[1] N. Dadkhah and B. Mettler, "Survey of motion planning literature in the presence of uncertainty: considerations for uav guidance," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 233–246, 2012.

[2] X. Hou and R. Mahony, "Dynamic kinesthetic boundary for haptic teleoperation of vtol aerial robots in complex environments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1–12, 2015.

[3] M. Furci, R. Naldi, S. Karaman, and L. Marconi, "A combined planning and control strategy for mobile robots navigation in populated environments," in *IEEE Conference on Decision and Control*, 2015.

[4] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.

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

[6] W. Van Loock, G. Pipeleers, and J. Swevers, "Time-optimal quadrotor flight," in *European Control Conference*, 2013, pp. 1788–1792.

[7] A. Bry, C. Richter, A. Bachrach, and N. Roy, "Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments," *The International Journal of Robotics Research*, 2015.

[8] E. Koyuncu and G. Inalhan, "A probabilistic b-spline motion planning algorithm for unmanned helicopters flying in dense 3d environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 815–821.

[9] P. M. Bouffard and S. L. Waslander, "A hybrid randomized/nonlinear programming technique for small aerial vehicle trajectory planning in 3d," *Planning, Perception and Navigation for Intelligent Vehicles*, vol. 63, 2009.

[10] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "Direct method based control system for an autonomous quadrotor," *Journal of Intelligent & Robotic Systems*, vol. 60, pp. 285–316, 2010.

[11] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1917–1922.

[12] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," in *IFAC world congress*, vol. 15, 2002, p. 310.

[13] J. Hauser and A. Saccon, "A barrier function method for the optimization of trajectory functionals with constraints," in *IEEE Conference on Decision and Control*, 2006, pp. 864–869.

[14] A. J. Hausler, A. Saccon, A. P. Aguiar, J. Hauser, and A. M. Pascoal, "Energy-optimal motion planning for multiple robotic vehicles with collision avoidance," *IEEE Transactions on Control Systems Technology*, 2015.

[15] A. Rucco, A. P. Aguiar, and J. Hauser, "A virtual target approach for trajectory optimization of a general class of constrained vehicles," in *IEEE Conference on Decision and Control*, 2015.

[16] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated vtol vehicles: A review of basic control design ideas and principles," *IEEE Control Systems*, vol. 33, no. 1, pp. 61–75, 2013.