

A Prototype of an Autonomous Controller for a Quadrotor UAV

Ian D. Cowling, Oleg A. Yakimenko, James F. Whidborne, and Alastair K. Cooke

Abstract— The paper proposes a complete real-time control algorithm for autonomous collision-free operations of the quadrotor UAV. As opposed to fixed wing vehicles the quadrotor is a small agile vehicle which might be more suitable for the variety of specific applications including search and rescue, surveillance and remote inspection. The developed control system incorporates both trajectory planning and path following. Using a differential flatness property the trajectory planning is posed as a constrained optimization problem in the output space (as opposed to the control space), which simplifies the problem. The trajectory and speed profile are parameterized to reduce the problem to a finite dimensional problem. To optimize the speed profile independently of the trajectory a virtual argument is used as opposed to time. A path following portion of the proposed algorithm uses a standard linear multi-variable control technique. The paper presents the results of simulations to demonstrate the suitability of the proposed control algorithm.

Key Words: quadrotor, trajectory planning, differential flatness, direct method, path following, LQR controller.

I. INTRODUCTION

Recently unmanned air vehicles (UAV's) have attracted considerable interest for a wide variety of different applications. While fixed wing vehicles have had extensive applications for military and meteorological purposes due to their range, speed and flight duration, rotorcraft vehicles are considered to be more preferable for surveillance, precise delivery and some other missions requiring agility and accuracy.

The quadrotor is a small agile vehicle controlled by the rotational speed of the four rotors. It benefits from having very few constraints on motion and an ability to carry a high payload. Furthermore, with the use of ducted fans instead of prop rotors it is safe for internal flights. The low cost and simplicity mean the quadrotor provides an excellent testing ground for application of advanced control techniques on UAV's.

In order to achieve full autonomy, a controller has to incorporate both trajectory planning and path following. Usually, these two problems are treated separately [1]. To date, several techniques based on the ideas of the direct method of calculus of variations have been developed for off-line optimization [2]-[19], [34]. However real-time applications require another approach [20]-[28]. Differential flatness is a dynamic property of some systems [29]. Exploiting the differential flatness of the vehicle dynamics provides a much more effective and robust way to obtain the quasi-optimal trajectories on board. The reason for that is that the differential flatness enables the optimization to occur within the output space as opposed to the control space. This technique has already been considered for space and air vehicles [30] and applied to a helicopter [31]. Differential flatness has also been considered with respect to the quadrotor to achieve a convergent tracking controller [32]. Path following is also a well-established problem. Once the reference trajectory $\mathbf{x}_{ref}$ and the nominal control $\mathbf{u}_{ref}$ are found, different approaches may be implemented to follow them [33], [35] and [41].

In this paper, the authors address the trajectory planning problem and apply a linear controller to achieve full autonomy of the prototype control system. First, Section II introduces the six degree of freedom (6DoF) model of the quadrotor. Section III discusses the general architecture of the autonomous control system prototype. The formulation of the optimization problem applied to the differentially flat systems is discussed in Section IV, followed by the detailed numerical optimization routine in Section V. The results of several simulations validating the developed controller are presented in Section VI. The paper ends with the conclusions.

II. QUADROTOR'S DYNAMICS AND SIMPLIFIED MODEL

By design, the quadrotor is only controlled by independently varying the speed of the four rotors (see Fig.1).

Let v_i and τ_i be the torque and thrust for i th rotor respectively ($i = 1, \dots, 4$) (these values are normalized with the moment of inertia and mass, correspondingly). Denoting the distance of the rotor from the center of mass by l , we now can introduce a set of four control inputs, u_i , as functions of normalized individual thrusts and torques as

Manuscript received September 30, 2006.

I. D. Cowling, J. F. Whidborne and A. K. Cooke are with the Department of Aerospace Sciences of Cranfield University, Bedfordshire, MK43 0AL, UK (emails: i.d.cowling {j.f.whidborne; a.cooke}@cranfield.ac.uk).

O. A. Yakimenko is with the Department of Mechanical and Astronautical Engineering of Naval Postgraduate School, Monterey, CA 93943-5107 USA (email: oayakime@nps.edu).

follows [37].

The total thrust is given by

$$u_1 = (\tau_1 + \tau_2 + \tau_3 + \tau_4); \quad (1)$$

a roll moment is achieved by varying the left (#4) and right (#3) rotor speeds

$$u_2 = l(\tau_4 - \tau_3); \quad (2)$$

a pitch moment is produced by varying the ratio of the front (#1) and back (#2) rotor speeds

$$u_3 = l(\tau_1 - \tau_2); \quad (3)$$

and finally, a yaw moment is obtained from the torque resulting from the subtracting counterclockwise (front and back) from the clockwise (left and right) speeds

$$u_4 = (v_3 + v_4 - v_1 - v_2). \quad (4)$$

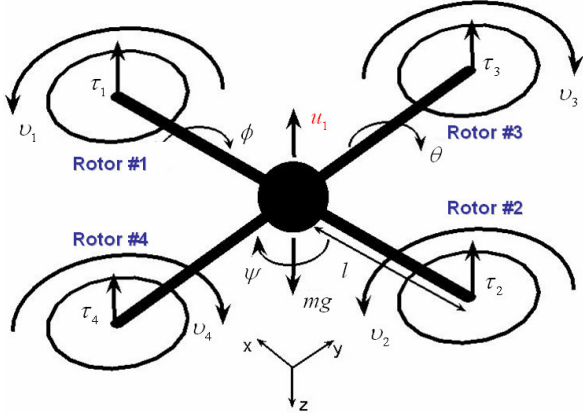


Figure 1. Quadrotor schematic.

The control vector, $\mathbf{u}$, is therefore defined as

$$\mathbf{u} = [u_1, u_2, u_3, u_4]^T, \quad (5)$$

where the last three control inputs govern the rotational dynamics directly:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix}. \quad (6)$$

(here p , q and r denote the body angular rates).

Let us further introduce a twelve-state vector

$$\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T, \quad (7)$$

where x , y and z are the translational positions (in the North-East-Down coordinate frame as shown on Fig.1), and ϕ , θ and ψ are the roll, pitch and yaw, respectively.

Typically, the equations of motion are determined using the standard aeronautical rotational matrix $\mathbf{R}_{xyz}$ ($\mathbf{R}_{\phi\theta\psi}$) as discussed in [38]. However, in this particular application the rotational matrix $\mathbf{R}_{zyx}$ ($\mathbf{R}_{\psi\theta\phi}$) is used as suggested in [36], [39], and [41]. This results in a decoupling of the equations of motion as will be shown shortly. Using the matrix $\mathbf{R}_{zyx}$, the Euler attitude rates, $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$, are related to the body angular rates by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ \sin \psi \tan \phi & \cos \psi \tan \phi & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (8)$$

If u_4 is used to keep the yaw angle, ψ , at zero and assuming that the pitch angle, θ , is small [39], then combining (6) and (8) yields

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix}. \quad (9)$$

Now, the translational equations of motion are

$$\ddot{x} = -u_1 \cos \phi \sin \theta, \quad (10)$$

$$\ddot{y} = u_1 \sin \phi, \quad (11)$$

and

$$\ddot{z} = g - u_1 \cos \phi \cos \theta. \quad (12)$$

Therefore, we can write the complete set of equations for the state vector (7) as

$$\dot{\mathbf{x}} = \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ -u_1 \cos \phi \sin \theta \\ u_1 \sin \phi \\ g - u_1 \cos \phi \cos \theta \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \mathbf{g}(\mathbf{x}, \mathbf{u}). \quad (13)$$

In order to address the differential flatness of the quadrotor's dynamics (to be discussed in Section IV.C) it is also necessary to define the four outputs (since we have four controls). These outputs are chosen to be the translational positions, x , y , z , and the yaw angle, ψ (because it can be dynamically decoupled from the other states in case the control input u_4 is used to set the yaw angle to zero). The output vector, $\mathbf{y}$, is hence defined as

$$\mathbf{y} = \begin{bmatrix} x \\ y \\ z \\ \psi \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 5} & 0 & \mathbf{0}_{3 \times 3} \\ 0 & \mathbf{0}_{1 \times 5} & 1 & \mathbf{0}_{1 \times 3} \end{bmatrix} \mathbf{x} = \mathbf{C}\mathbf{x}. \quad (14)$$

III. PATH FOLLOWING WITH THE LQR CONTROLLER

Before we proceed any further let us develop a general architecture of the controller. It is suggested that the controller consist of two parts (Fig.2).

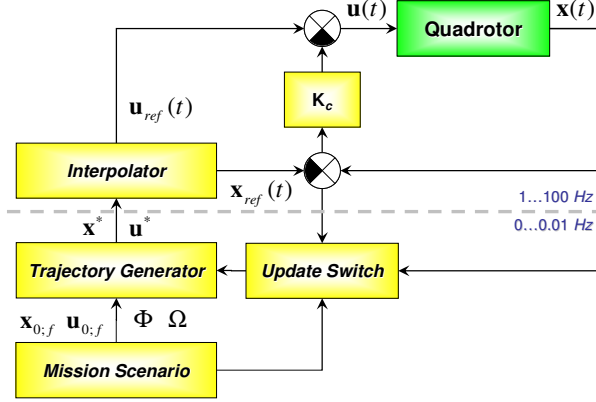


Figure 2. General architecture of the quadrotor's controller.

Based on the particular mission scenario, the trajectory generator produces an optimal/quasioptimal/feasible trajectory. In terms of mathematics it means that based on the boundary (initial and final) conditions, imposed onto the state $\mathbf{x}$ and control $\mathbf{u}$, as well as mission performance index Φ to be optimized, and a set of some constraints Ω , it is desired to develop the best possible trajectory. Ideally, we of course would like to have an optimal trajectory, but if it takes too much time to produce it or if this trajectory cannot be found at all we would rather have an algorithm allowing to find an “almost” optimal (quasi-optimal) or even simply feasible trajectory, but in real-time.

Real-time trajectory generation capability allows regeneration of the trajectory during the mission if for instance the objectives for the mission change or the discrepancy between the current state and the suggested path becomes too large (because of disturbances and an imperfection of the controller). If such an event occurs, the update switch (see Fig.2) forces the trajectory generator to calculate a new quasi-optimal trajectory passing through the current state to count it as the new vector of initial conditions.

It is suggested that once the reasonable reference trajectory to fulfill the mission, $\mathbf{x}^*$, and the corresponding nominal control, $\mathbf{u}^*$, are found, we may use an LQR to track it in the presence of disturbances [41]. In this case the interpolator produces samples of the reference trajectory at the desired (high frequency) rate.

A. Formulation of the LQR Problem

For a linear state-space model of the plant dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (15)$$

a control input

$$\mathbf{u}(t) = -\mathbf{K}_c \mathbf{x}(t) \quad (16)$$

is determined such that the closed loop system

$$\dot{\mathbf{x}} = [\mathbf{A} - \mathbf{B}\mathbf{K}_c] \mathbf{x}(t) \quad (17)$$

is stable and a performance measure where

$$J = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (18)$$

is minimized ($\mathbf{Q}$ and $\mathbf{R}$ are the weighting matrices).

Now, assuming a time-dependent reference trajectory $\mathbf{x}_{ref}(t)$, the LQR control can be applied as a trajectory follower to minimize small errors between the full measured state $\mathbf{x}$ and the reference state $\mathbf{x}_{ref}$, such that the applied control becomes

$$\mathbf{u}(t) = \mathbf{u}_{ref}(t) - \mathbf{K}_c (\mathbf{x}(t) - \mathbf{x}_{ref}(t)). \quad (19)$$

We will compute the gain matrix $\mathbf{K}_c$ using a linearized version of the quadrotor's model (13) and use this control candidate later in simulations for a non-linear model.

B. Stability Analysis

The control gains $\mathbf{K}_c$ were designed with the plant linearized at hover with the weighting matrices

$$\mathbf{Q} = 10^{-5} \mathbf{I}_{12 \times 12} \text{ and } \mathbf{R} = \text{diag}(10^{-5}, 10^8, 10^8, 10^8) \quad (20)$$

ensuring that the actuator constraints are maintained [43].

However, following a spatial trajectory obviously violates the hover conditions. A simplified analysis was performed within [41], to determine an envelope of stable operation. As shown in [41], to assure linearized time-invariant stability, an extra constraint, which maintains angles ϕ and θ within the set ζ_{st} , is added to the optimization, within the trajectory planner.

$$\zeta_{st} = \{\theta, \phi: \theta^2 + \phi^2 \leq r^2, r = 48^\circ\} \quad (21)$$

IV. TRAJECTORY OPTIMIZATION

Now that we have agreed on the general control scheme and introduced a simple LQR controller, we may proceed with a matter of finding the reference trajectory $\mathbf{x}_{ref}(t)$ and nominal control profile $\mathbf{u}_{ref}(t)$. This section deals with two formulations of the optimization problem: for a general case and for the specific case when the system's dynamics possess a differential flatness property. The next section takes advantage of a specific formulation and introduces the step-by-step optimization routine.

A. Problem Formulation in the Control Space

Typically, to determine the optimal (reference) trajectory, an optimization within the control space is performed subject to some constraints placed within the output space and the state space. These constraints include state constraints, actuator (control) constraints and obstacle avoidance constraints. For example, the problem can be posed as

$$\begin{aligned} \min_{\mathbf{u}(t) \in \mathbf{U} \subset \mathbb{R}^4} \Phi \quad & \text{for } t \in [0, t_f] \\ \text{s.t.: } \quad & \dot{\mathbf{y}} - \mathbf{C}\mathbf{g}(\mathbf{x}, \mathbf{u}) = 0 \\ & \dot{\mathbf{y}}_0 - \mathbf{C}\mathbf{g}(\mathbf{x}_0, \mathbf{u}_0) = 0 \\ & \dot{\mathbf{y}}_f - \mathbf{C}\mathbf{g}(\mathbf{x}_f, \mathbf{u}_f) = 0 \\ & \mathbf{c}(\mathbf{x}, \mathbf{u}) \leq 0 \end{aligned} \quad (22)$$

where Φ is the cost function, initial and final constraints are placed on the state in accordance with (13) and (14) at

$t = 0$ and $t = t_f$, respectfully, dynamic inequality constraints on the trajectory (for obstacle avoidance) and on the state and input (to avoid singularities and to provide constraints on the control signals) are expressed through the set of functions $\mathbf{c}(\mathbf{x}, \mathbf{u})$.

B. Cost Function

The cost function, Φ , is a quantitative measure of the optimality of the trajectory and can be approximated by the sum of the running costs and the terminal cost. Assuming that the running costs (fuel consumption) are proportional to the average velocity, the objective function can be defined as:

$$\Phi = (1-w) \frac{1}{t_f} \int_0^{t_f} \sqrt{P_1 \dot{x}^2 + P_2 \dot{y}^2 + P_3 \dot{z}^2} dt + w(t_f - T)^2, \quad (23)$$

where w , P_1 , P_2 , and P_3 are the weighting factors (for more flexibility, the last three of them are not necessarily equal to each other), and T is the predetermined time of arrival. Particularly, the case when $w=1$ and $T=0$ corresponds to the minimum-time problem; and the case when $w=0$, $P_1 = P_2 = P_3$ corresponds to the minimum-fuel problem. Therefore mathematically, the mission reassessment situation mentioned in Section III might involve changing four weighting factors and time of arrival T .

C. Differential Flatness

By definition the differential flatness is the expression of the state and control vectors in terms of the output vector [29]. For a system to be differentially flat and therefore possess a flat output it requires a set of variables such that [42]:

- the components of $\mathbf{y}$ are not differentially related over $\mathcal{R}$;
- every system variable may be expressed as a function of the output $\mathbf{y}$;
- conversely, every component of $\mathbf{y}$ may be expressed as a function of the system variables and of a finite number of their time derivatives.

The components of the control input, $\mathbf{u}$, can be easily expressed in terms of the states and their derivatives from (10-12) and (9)

$$\begin{aligned} u_1 &= \sqrt{\ddot{x}^2 + \ddot{y}^2 + (g - \ddot{z})^2} \\ u_2 &= \ddot{\phi} \\ u_3 &= \ddot{\theta} \\ u_4 &= \ddot{\psi} \end{aligned} \quad (24)$$

From (10)-(12) it also follows that the rotational part of the state vector can be expressed as a function of the output vector and its derivatives as

$$\theta = \arctan \frac{-\ddot{x}}{g - \ddot{z}} \quad (25)$$

and

$$\phi = \arcsin \frac{\ddot{y}}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + (g - \ddot{z})^2}}. \quad (26)$$

Singularities in (25) and (26) can only appear when $g = \ddot{z}$, in other words when the vehicle is in free fall. This can be avoided by constraining the input such that $u_1 > 0$ and the pitch and roll such that $\theta < 90^\circ$ and $\phi < 90^\circ$ (see (12)). These angles are outside the set ζ_{st} already defined in (21).

The derivatives of (25) and (26) yield

$$\dot{\theta} = -\frac{\ddot{x}(g - \ddot{z}) + \dot{x}\ddot{z}}{(g - \ddot{z})^2 + \dot{x}^2} \quad (27)$$

and

$$\dot{\phi} = \frac{(\dot{x}^2 + (g - \ddot{z})^2)\ddot{y} - (\ddot{x}\ddot{x} - (g - \ddot{z})\ddot{z})\ddot{y}}{(\dot{x}^2 + \ddot{y}^2 + (g - \ddot{z})^2)\sqrt{\dot{x}^2 + (g - \ddot{z})^2}}. \quad (28)$$

Equations (27) and (28) can be differentiated one more time to be substituted into (24).

To summarize, the state vector $\mathbf{x}$ and the control vector $\mathbf{u}$ can be both expressed via (derivatives of) the output vector $\mathbf{y}$, i.e.

$$\mathbf{x} = \mathbf{h}_1(\mathbf{y}), \quad \mathbf{u} = \mathbf{h}_2(\mathbf{y}). \quad (29)$$

Therefore, the general optimization problem can now be reformulated.

D. Problem Formulation in the Output Space

Having the differentially flat equations, the problem can be reposed to allow optimization to occur within the output space as opposed to the control space. This is truly beneficial because the constraints arising for example from obstacle avoidance occur in the output space, hence the computation time for constraint handling is reduced drastically. As opposed to (22) the problem can now be posed as follows:

$$\begin{aligned} \min_{\mathbf{y}(t) \in \mathbf{Y} \subset \mathcal{R}^4} \quad & \Phi \quad \text{for } t \in [0, t_f] \\ \text{s.t.:} \quad & \dot{\mathbf{y}}_0 - \mathbf{g}^*(\mathbf{y}(0)) = 0 \\ & \dot{\mathbf{y}}_f - \mathbf{g}^*(\mathbf{y}(t_f)) = 0 \\ & \mathbf{c}^*(\mathbf{y}) \leq 0 \end{aligned} \quad (30)$$

In this expression $\mathbf{g}^*(\mathbf{y}) = \mathbf{Cg}(\mathbf{x}, \mathbf{u}) = \mathbf{Cg}(\mathbf{h}_1(\mathbf{y}), \mathbf{h}_2(\mathbf{y}))$ and $\mathbf{c}^*(\mathbf{y}) = \mathbf{c}(\mathbf{x}, \mathbf{u}) = \mathbf{c}(\mathbf{h}_1(\mathbf{y}), \mathbf{h}_2(\mathbf{y}))$ profit from the differential flatness (29). Using a suitable parameterization introduced in the next section this problem can be entered into MATLAB and solved using the optimization toolbox function *fmincon*.

V. PARAMETERIZATION

In order to reduce the dimension of the problem to a finite amount, it is suggested that the three translational outputs (x , y and z) be parameterized (the fourth output, the yaw angle ψ , is assumed to be zero).

There are numerous alternatives for doing this [20], such

as approximations using the simple monomials (elementary polynomials) [22], [23], [26]-[28], [43], Chebyshev polynomials [18], [44], Laguerre polynomials [45], etc. Every technique, however, can be expressed as a finite series involving a product of a free variable a_k and a basis function Γ_k

$$P(t) = \sum_{k=0}^M a_k \Gamma_k(t), \quad (31)$$

where M is the order of parameterization (approximation).

For example, the monomial basis function can be expressed as

$$\Gamma_k = t^k \text{ or } \Gamma_k = \frac{1}{k!} t^k \quad (32)$$

(the last one arises from the Taylor series expansion), Chebyshev polynomials (orthogonal over the interval $[-1;1]$) can be represented either geometrically as

$$\Gamma_k(t) = \cos(k \arccos(t)) \quad (33)$$

or defined by a recurrence relationship [46]

$$\begin{aligned} \Gamma_0(t) &= 1 \\ \Gamma_1(t) &= t \end{aligned} \quad (34)$$

$$\Gamma_{k+1}(t) = 2t\Gamma_k(t) - \Gamma_{k-1}(t), \quad k \geq 1$$

and also orthogonal (over $[-1;1]$) Laguerre polynomials can be derived by another recurrence relationship [46]

$$\begin{aligned} \Gamma_0(t) &= 1 \\ \Gamma_1(t) &= 1-t \end{aligned} \quad (35)$$

$$(k+1)\Gamma_{k+1}(t) = (2k+1-t)\Gamma_k(t) - k\Gamma_{k-1}(t), \quad k \geq 1$$

Obviously, the choice of a particular basis function depends on the properties of vehicle's dynamics and specific application. This choice can also affect the numerical robustness. In our case the search space reduces to $\mathbb{R}^{3(M+1)}$ ($M+1$ free variables a_k , $k=0, \dots, M$ per each Cartesian coordinate).

A. Satisfying Boundary Conditions

Let us show now how to choose the order of approximation M . Suppose we have to satisfy up to the second derivatives of output $\mathbf{y}$ at both ends of the trajectory, i.e. $x_{0,f}^{(n)}$, $y_{0,f}^{(n)}$, and $z_{0,f}^{(n)}$, $n=1, \dots, 2$ are given. That means that for each coordinate we have six constraints and therefore the order of approximation should be at least $M=5$ (otherwise we will have less than six free variables a_k , and not be able to satisfy all the constraints).

If $M=5$ we can determine all coefficients a_k analytically and therefore satisfy all boundary conditions in (30) a priori. As an example let us consider the case of monomials. However we will not use any of the form (32)-(35), but employ another form to simplify analytical formulas. We start from approximating the second derivative for each coordinate with a third-order polynomial of the first form of (32), e.g.

$$\ddot{x}(t) = a_{x2} + a_{x3}t + a_{x4}t^2 + a_{x5}t^3 = \sum_{k=2}^5 a_{xk}t^{k-2} \quad (36)$$

(in the same manner we define $\ddot{y}(t)$ and $\ddot{z}(t)$).

Integrating (36) twice yields the following 4th- and 5th-order polynomials for the first derivative and for the x coordinate itself:

$$\dot{x}(t) = \sum_{k=1}^5 \frac{a_{xk}t^{k-1}}{\max(1, k-1)}, \quad x(t) = \sum_{k=0}^5 \frac{a_{xk}t^k}{\max(1, k(k-1))}. \quad (37)$$

Now, all six coefficients a_{xk} , $k=0, \dots, 5$ can be defined from the following linear matrix equation:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & t_f & \frac{1}{2}t_f^2 & \frac{1}{6}t_f^3 & \frac{1}{12}t_f^4 & \frac{1}{20}t_f^5 \\ 0 & 1 & t_f & \frac{1}{2}t_f^2 & \frac{1}{3}t_f^3 & \frac{1}{4}t_f^4 \\ 0 & 0 & 1 & t_f & t_f^2 & t_f^3 \end{bmatrix} \begin{bmatrix} a_{x0} \\ a_{x1} \\ a_{x2} \\ a_{x3} \\ a_{x4} \\ a_{x5} \end{bmatrix} = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \\ x_f \\ \dot{x}_f \\ \ddot{x}_f \end{bmatrix} \quad (38)$$

(again, in the same manner we determine a_{yk} and a_{zk}).

In this particular case ($M=5$) the only varied parameter is t_f (since all coefficients are determined from the boundary conditions). Figure 3a demonstrates what happens when t_f is varied. But what if in addition to t_f some other coefficients can be varied too? What if say the second derivative of y at the final point, $\ddot{y}_f$, is not defined (meaning it becomes an additional varied parameter)? Figure 3b provides with an example of such enhanced flexibility of the reference functions.

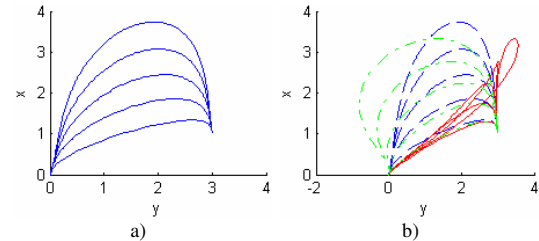


Figure 3. Varying "free" parameters of the reference functions: $t_f = \text{var}$ (a), $t_f = \text{var}$ & $\ddot{y}_f = \text{var}$ (b).

To summarize, if an additional flexibility is needed – we can always increase the order of approximation (36) and use the higher-order (artificially introduced) derivatives at any (or both) end as additional varied parameters. In our particular case 7th-order polynomials (rather than 5th-order) have been used. The initial and terminal state, first and second derivatives were considered as constraints to be satisfied and the third derivatives at both ends of the trajectory became free variables along with the virtual arc length.

B. Separating Trajectory from Speed Profile

Having time, t , as an argument in (31) means that not only do we define the trajectory itself, but we also preset a single speed profile along the trajectory. This is because the speed is unambiguously related to the Cartesian coordinates via

$$V(t) = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2}. \quad (39)$$

The only way to separate the trajectory from the speed profile is to use some abstract argument τ that relates to time via the variable speed factor $\lambda(\tau) = d\tau/dt$ [21]. In this case we obtain an authority to vary the speed profile along the predetermined trajectory via varying the speed factor $\lambda(\tau)$

$$V(\tau) = \lambda(\tau) \sqrt{x'(\tau)^2 + y'(\tau)^2 + z'(\tau)^2}. \quad (40)$$

The speed profile $V(\tau)$ can now be also approximated with say the same-order polynomial

$$V(\tau) = \sum_{k=0}^5 \frac{a_{vk} \tau^k}{\max(1, k(k-1))} \quad (41)$$

(in this case we will have three variable parameters since the second derivatives of V are not defined by the boundary conditions).

C. Resolving Inverse Dynamics

The complete computational algorithm looks as follows. We start from dividing the virtual arc τ_f (varied parameter) into $N-1$ equal pieces $\Delta\tau = \tau_f(N-1)^{-1}$ so that we have N equidistant (along τ_f) nodes $j=1, \dots, N$. All the states and controls at the first point $j=1$ (corresponding to $\tau_1 \equiv \tau_0 = 0$) are defined.

Now, for each of the subsequent $N-1$ nodes $j=2, \dots, N$ we do the following. We compute the current value of three Cartesian coordinates and speed using the corresponding polynomials: $x_j = P_x(\tau_j)$, $y_j = P_y(\tau_j)$, $z_j = P_z(\tau_j)$, and $V_j = P_v(\tau_j)$. Then, using the obvious relationship we compute the time passed since the last sample

$$\Delta t_{j-1} = 2 \frac{\sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2 + (z_j - z_{j-1})^2}}{V_j + V_{j-1}}, \quad (42)$$

(the current time then equals to $t_j = t_{j-1} + \Delta t_{j-1}$ ($t_1 \equiv t_0 = 0$)), and therefore the current value of the speed factor

$$\lambda_j = \frac{\Delta\tau}{\Delta t_{j-1}}. \quad (43)$$

Next, we convert τ (prime) derivatives obtained directly from analytically defined $P_x(\tau)$, $P_y(\tau)$, $P_z(\tau)$, and $P_v(\tau)$ to time derivatives using obvious relations

$$\dot{\xi} = \lambda \xi', \quad \ddot{\xi} = \lambda(\lambda' \xi' + \lambda \xi''), \text{ and so on} \quad (44)$$

(by the way, we use the inverse of these relations to convert the initial conditions of the states from time derivatives to virtual arc derivatives).

Having those we use (24)-(28) to compute the controls and the remaining states.

VI. SIMULATION WITH FULL MODEL

To validate the control strategy a full simulation model of the quadrotor has been developed [36]. The full SIMULINK model incorporates experimental data and theoretical relations. This model is used to test the control algorithm to fly three missions discussed next.

A. Missions and Constraints

To test the control scheme, three missions have been created to test the ability and robustness of the controller. As discussed the cost function can be expressed as a function of total flight time (time optimal), the total distance traveled (fuel optimal) or a combination of the two. For all of the following missions the cost function is set up as an optimal time problem. All missions involve starting from hover at $(0, 0, 0)m$ and moving to a certain destination within the preset time with a sampling rate of 10Hz:

- The first mission is reduced to a simple vertical flight of 7m finishing in a hover at the destination;
- The second mission involves navigation around an obstacle to reach a destination at $(6, 0, 0)m$. The obstacle is modeled as a sphere of a 1m radius centered at $(3, 0, 0)m$;
- The third mission assumes a horizontal flight to $(10, 0, 0)m$ followed by a decent down a mineshaft of a 2m radius to its bottom $(10, 0, -5)m$.

The dynamic constraints consist of input constraints

$$0.5 \leq u_i \leq 5.4, \quad -0.5 \leq u_i \leq 0.5, \quad i=2,3,4, \quad (45)$$

obstacle avoidance constraints and stability constraint (21). These input constraints are reasonable approximations of the actuator outputs as it can be assumed u_i for $i=2,3,4$, is small relative to u_1 .

B. Modeling Disturbances

To test the robustness of the controller, disturbances have been introduced to the full quadrotor model. The likely disturbances acting on the quadrotor are wind and noise. Wind can be modeled as a drift in the translational position, more precisely a velocity of 0.1m/s along the x axis (this can further be improved with a gusting model of random noise in the same direction). To simulate inaccuracies within the rotor, a random noise was also added to the input signal.

C. Results

All aforementioned scenarios were tested using different weighting factors in (23). Below we only show one set of results per each scenario with $w=1$ and $T=0$ corresponding to the minimum-time problem. It should be noted that if the only type of the problems to consider be the time-minimum

problems, then instead of defining the reference polynomial for speed (41) we would establish a bang-bang control profile for u_1 and then integrate the speed equation (as it was done for other applications, for example in [23] and [26]). That means rather than having two varied coefficients in (41) we would have several varied switching points to play with. However, having (41) increases the class of the cost functions to be considered (we simply do not care whether the control enters the Hamiltonian of the system (13) linearly). Therefore, we are showing the minimum-time problem results to show that the developed algorithms work here too despite of the fact that we intentionally replaced an optimal bang-bang control with the quasi-optimal polynomial control that can match the bang-bang control to a certain extent only.

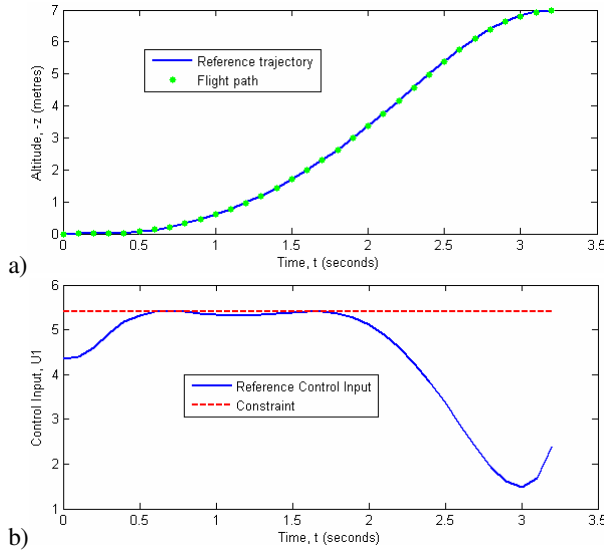


Figure 4. The altitude (a) and thrust (b) profiles for the vertical flight.

The solution to the first problem (minimum-time vertical climb) is well known [47]: the vehicle applies maximum thrust followed by minimum thrust to slow down. The analytical solution for this problem is the application of maximum thrust for approximately 1.44 seconds followed by zero thrust for 0.36 seconds (of course this solution was not obtained for the full model (13) under (30) but it at least gives some feelings of what final time to expect). Figure 4a shows the altitude against time until the destination is reached at the final time of 3.26 seconds. Figure 4b shows the first control input over this time. As seen the thrust is a polynomial approximation of the maximum constraint followed by a short reduction in thrust. It should be noted though that while time t_f is 1.5 times greater than the optimal one (obtained for the simplified dynamics) all boundary conditions (including those imposed on the rotor fans' dynamics) are satisfied, the reference trajectory and nominal control are feasible and (which is probably the most important) the solution can be obtained in the real-

time.

The path following part of the algorithm works fairly well following the quasi-optimal trajectory with a high accuracy (Fig.4a). One of the main reasons for that is that all constraints are carefully taken into account and therefore the LQR controller does not need to “fight” inaccuracies in the model (which we would inevitably encounter if the results of the truly optimal solution, but for the simplified model, were employed).

For the second mission (obstacle avoidance), as the optimal trajectory is measured by the flight time, it follows, that the optimal trajectory is the shortest path (and the thrust control looks pretty much the same as in the previous task). This can be seen by the trajectory passing along the surface of the trajectory (Fig.5). The flight time for this mission is 3.7 seconds. As seen the vehicle tracks the reference trajectory in this case fairly well too despite the presence of the wind and other disturbances.

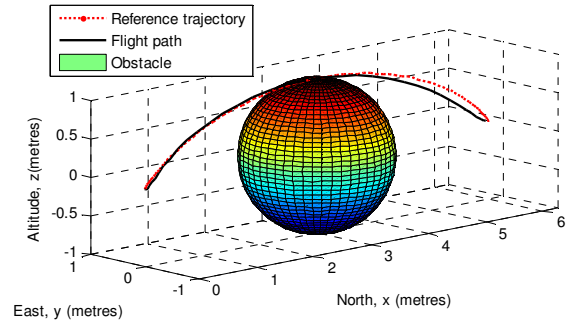


Figure 5. Obstacle avoidance.

The final mission is the mineshaft descent mission. The vehicle starts at hover and flies to the top of and down a mineshaft (Fig.6). The flight time for this mission is 8.6 seconds.

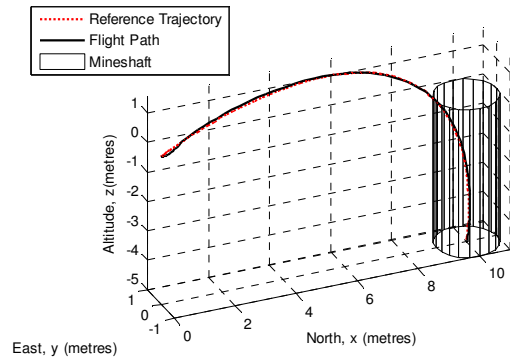


Figure 6. Mineshaft mission.

VII. CONCLUSIONS

This paper presented a quasioptimal trajectory planner (benefiting from the differential flatness and the direct method) with a simple LQR path following controller. This scheme has been validated using a full dynamic model of the quadrotor. Simulations prove the feasibility of considered approach and show that the quadrotor's model

follows the optimal trajectories generated in real-time despite wind and other perturbations fairly well. Future work will look at extending the developed controller to make it more robust and capable of handling collision-free flight in presence of multiple vehicles.

REFERENCES

- [1] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using Mixed Integer Linear Programming," in *Proc. Amer. Contr. Conf.*, Anchorage, AK, May 8-10, 2002.
- [2] C. R. Hargraves and S. W. Paris, "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, vol.10, no.4, 1987, pp.338-342.
- [3] S. H. Lane and R. F. Stengel, "Flight Control Design Using Nonlinear Inverse Dynamics," *Automatica*, vol.24, no.4, 1988, pp.471-483.
- [4] O. Von Stryk and R. Bulirsch, "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, vol.37, 1992, pp.357-373.
- [5] E. Sentoh and A. E. Bryson, "Inverse and Optimal Control for Desired Outputs," *Journal of Guidance, Control, and Dynamics*, vol.15, no.3, 1992, pp.687-691.
- [6] P. J. Enright and B. A. Conway, "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, vol.15, no.5, 1992, pp.994-1002.
- [7] P. Lu, "Inverse Dynamics Approach to Trajectory Optimization for an Aerospace Plane," *Journal of Guidance, Control, and Dynamics*, vol.16, no.4, 1993, pp.726-732.
- [8] H. Seywald, "Trajectory Optimization Based on Differential Inclusions," *Journal of Guidance, Control, and Dynamics*, vol.17, no.3, 1994, pp.480-487.
- [9] J. Elnagar, M. A. Kazemi and M. Razzaghi, "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, vol. 40, no.10, 1995, pp.1793-1796.
- [10] R. Kumar and H. Seywald, "Should Controls Be Eliminated While Solving Optimal Control Problems via Direct Methods?" *Journal of Guidance, Control, and Dynamics*, vol.19, no.2, 1996, pp.418-423.
- [11] R. Kumar and H. Seywald, "Dense-Sparse Discretization for Optimization and Real-Time Guidance," *Journal of Guidance, Control, and Dynamics*, vol.19, no.2, 1996, pp.501-503.
- [12] A. L. Herman and B. A. Conway, "Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, vol.19, no.3, 1996, pp.592-599.
- [13] D. G. Hull, "Conversion of Optimal Control Problems into Parameter Optimization Problems," *Journal of Guidance, Control, and Dynamics*, vol.20, no.1, 1997, pp.57-60.
- [14] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, vol.21, no.2, 1998, pp.193-207.
- [15] B. A. Conway and K. M. Larson, "Collocation Versus Differential Inclusion in Direct Optimization," *Journal of Guidance, Control, and Dynamics*, vol.21, no.5, 1998, pp.780-785.
- [16] F. Fahroo and I. M. Ross, "Computational Optimal Control by Spectral Collocation with Differential Inclusion," in *Proc. of the 1999 Flight Mechanics Symposium*, NASA-Goddard Space Flight Center, Greenbelt, MD, May 18-20, 1999, pp.185-200.
- [17] F. Fahroo and I. M. Ross, "A Second Look at Approximating Differential Inclusions," *Journal of Guidance, Control, and Dynamics*, vol.24, no.1, 2001, pp.131-133.
- [18] F. Fahroo and I. M. Ross, "Direct Trajectory Optimization by a Chebyshev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, vol.25, no.1, 2002, pp.160-166.
- [19] F. Fahroo and I. M. Ross, "Pseudospectral Knotting Methods for Solving Nonsmooth Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, vol.27, no.3, 2004, pp.397-405.
- [20] V. T. Taranenko, *Experience on Application of Ritz's, Poincare's, and Lyapunov's Methods in Solving of Flight Dynamics Problems*. Air Force Engineering Academy Press, Moscow, 1968 (in Russian).
- [21] V. T. Taranenko and V. G. Momdzh, *Direct Variational Method in Boundary Problems of Flight Dynamics*. Mashinostroenie Press, Moscow, 1968 (in Russian).
- [22] A. I. Neljubov, *Mathematical Methods of Calculation of Combat, Takeoff/Climb, and Landing Approach Manoeuvres for the Aircraft with 2-D Thrust Vectoring. Flight Characteristics and Combat Maneuvering of Manned Vehicles*. Air Force Engineering Academy Press, Moscow, 1968 (in Russian).
- [23] O. Yakimenko, "Direct Method for Rapid Prototyping of Near-Optimal Aircraft Trajectories," *Journal of Guidance, Control, and Dynamics*, vol.23, no.5, 2000, pp.865-875.
- [24] D. V. Alekhin and O. A. Yakimenko, "Synthesis of Optimization Algorithm for Planned-Route Flight Trajectory by Direct Variational Method," *Journal of Computer and Systems Sciences International*, vol.38, no.4, 1999, pp.650-666.
- [25] V. N. Dobrokhodov and O. A. Yakimenko, "Synthesis of Trajectory Control Algorithms at the Stage of Rendezvous of an Airplane with a Maneuvering Object," *Journal of Computer and Systems Sciences International*, vol.38, no.2, 1999, pp.262-277.
- [26] R. Bevilacqua, O. Yakimenko and M. Romano, "On-Line Generation of Quasi-Optimal Docking Trajectories," in *Proc. of the 7th International Conference On Dynamics and Control of Systems and Structures in Space (DCSSS)*, Greenwich, London, England, July 16-20, 2006.
- [27] B. Eikenberry, O. Yakimenko and M. Romano, "A Vision Based Navigation among Multiple Flocking Robots: Modeling and Simulation," in *Proc. of AIAA Modeling and Simulation Technologies Conference*, Keystone, CO, August 21-24, 2006.
- [28] O. Yakimenko, "Direct Method for Real-Time Prototyping of Optimal Control," in *Proc. of the International Conference Control 2006*, Glasgow, Scotland, August 30 - September 11, 2006.
- [29] M. Fleiss, J. Levine, Ph. Martin and P. Rouchon, "Sur les systèmes non linéaires différentiellement plats," in *C.R. Acad. Sci.*, Paris, vol.315, série I, 1992, pp.619-624.
- [30] P. Martin, S. Devasia and B. Paden, "A different look at output tracking: control of a VTOL aircraft," in *Proc. 33rd IEEE Conf. Decision Contr.*, Lake Buena Vista, FL, December 14-16, 1994.
- [31] T. J. Koo and S. Sastry, "Differential Flatness Based Full Authority Helicopter Design," in *Proc. 38th IEEE Conf. Decision Contr.*, Phoenix, AZ, December 7-10, 1999.
- [32] B. J. Driessen, and A.L. Robin, "A Globally Convergent Tracking Controller for the X4 Flyer Rotor Craft for Reference Trajectories with Positive Thrust," *Robotica*, part 4, 2004, pp.375-388.
- [33] A. Bouabdallah, A. Noth and R. Siegwart, "PID versus LQ control techniques applied to an indoor micro quadrotor," in *Proc. IEEE Conf. on Intel. Robot. Systems*, Sendai, Japan, September 28 - October 2, 2004.
- [34] W. A. Kamal, D.-W. Gu and I. Postlethwaite, "MILP and its application in flight path planning," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, July 3-8, 2005.
- [35] I. Kaminer, O. Yakimenko and A. Pascoal, "On Path Generation, Trajectory Tracking and Coordinated Control for Time-Critical Missions of Multiple UAVs," in *Proc. of IEEE American Control Conference*, Minneapolis, MN, June 14-16, 2006.
- [36] I. D. Cowling, J.F. Whidborne and A.K. Cooke, "MBPC for Autonomous Operation of a Quadrotor Air Vehicle," in *Proc 21st International UAV Systems Conf.*, Bristol, UK, April 3-5, 2006.
- [37] A. K. Cooke, *Helicopter test and evaluation*. Blackwell and QinetiQ, Oxford, England, 2002.
- [38] M. V. Cook, *Flight Dynamics Principles*. Butterworth Heinemann, Oxford, England, 1997.
- [39] P. Castillo, A. Dzul and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Transactions on Control Systems Technology*, vol.12, no.4, 2004, pp.510-516.
- [40] E. Altug, J. P. Ostrowski and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA2002)*, Washington, DC, May 11-15, 2002.
- [41] I. D. Cowling, J. F. Whidborne and A. K. Cooke, "Optimal trajectory planning and LQR control for a quadrotor UAV," in *Proc. of the International Conference Control 2006*, Glasgow, Scotland, August 30 - September 11, 2006.
- [42] A. Chelouah, "Extensions of differential flat fields and Liouvillian systems," in *Proc. 36th IEEE Conf. Decision Contr.*, San Diego, CA, December 10-12, 1997.
- [43] M. J. Nieuwstadt and R. M. Murray, "Approximate trajectory generation for differentially flat systems with zero dynamics," in *Proc. 34th IEEE Conf. Decision Contr.*, New Orleans, LA, December 13-15, 1995.
- [44] J. Vlassenbroeck, and R. Van Dooren, "A Chebyshev Technique for Solving Nonlinear Optimal Control Problems," *IEEE Trans. Autom. Control*, vol.33, no.4, 1988, pp.333-340.
- [45] M. Huzmezan, G. A. Dumont, W. A. Gough and S. Kovac, "Multivariable Laguerre-based indirect adaptive predictive control a reliable practical solution for process control," in *IASTED Modeling and Control Conf.*, Innsbruck, Austria, February 18-21, 2001.
- [46] R. J. Schilling, S. L. Harris, *Applied Numerical Methods for Engineers Using MATLAB and C*. Brooks/Cole Publishing Company, Pacific Grove, CA, 2000, pp.336-337.
- [47] J. T. Tou, *Modern Control Theory*, McGraw-Hill, New York, NY, 1964.



DSpace

<https://dspace.org/>

A prototype of an autonomous controller for a quadrotor UAV

Cowling, Ian D.; Yakimenko, Oleg A.; Whidborne, James F.; Cooke, Alastair K.

2007-02-02

IEEE

<http://doi.org/10.23919/ECC.2007.7068316>

Ian D. Cowling, Oleg A. Yakimenko, James F. Whidborne and Alastair K. Cooke. A prototype of an autonomous controller for a quadrotor UAV. Proceedings of the European Control Conference 2007, Kos, Greece, July 2-5, 2007, pp. 4001-4008