Flatness-Based Trajectory Planning/Replanning for a Quadrotor Unmanned Aerial Vehicle

ABBAS CHAMSEDDINE
YOUMIN ZHANG, Senior Member, IEEE
Concordia University

CAMILLE ALAIN RABBATH, Senior Member, IEEE
Defence Research and Development Canada

CÉDRIC JOIN
DIDIER THEILLIOL, Member, IEEE
Centre de Recherche en Automatique de Nancy

A flatness-based flight trajectory planning/replanning strategy is proposed for a quadrotor unmanned aerial vehicle (UAV). In the nominal situation (fault-free case), the objective is to drive the system from an initial position to a final one without hitting the actuator constraints while minimizing the total time of the mission or minimizing the total energy spent. When actuator faults occur, fault-tolerant control (FTC) is combined with trajectory replanning to change the reference trajectory in function of the remaining resources in the system. The approach employs differential flatness to express the control inputs to be applied in the function of the desired trajectories and formulates the trajectory planning/replanning problem as a constrained optimization problem.

Manuscript received March 18, 2010; revised October 13, 2010 and February 18, 2011; released for publication July 27, 2011.

IEEE Log No. T-AES/48/4/944176.

Refereeing of this contribution was handled by M. Blanke.

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Strategic Project Grant (STPGP 350889-07), the NSERC Discovery Project Grant, the NSERC Postdoctoral Fellowship program, the Defence Research and Development Canada (DRDC) Technology Investment Fund, and Numerica Technologies Inc., Quebec.

The ideas of flatness-based trajectory planning/replanning was originally developed during the Post-doctoral Fellowship of the first author at the Centre de Recherche en Automatique de Nancy (CRAN)—UMR CNRS 7039, Nancy, France in 2009 with the financial support of SIRASAS Project.

Authors' addresses: A. Chamseddine and Y. M. Zhang, Department of Mechanical & Industrial Engineering, Concordia University, 1455 Maisonneuve Blvd. W., Montreal, Quebec H3G 1M8, Canada, E-mail: (ymzhang@encs.concordia.ca); C. A. Rabbath, Defence Research and Development Canada (DRDC), 2459 Boul. Pie XI, Valcartier, Quebec G3J 1X5, Canada; C. Join, Centre de Recherche en Automatique de Nancy (CRAN)—CNRS UMR 7039 & NON-A—INRIA, University of Lorraine, BP 70239, 54506 Vandoeuvre Cedex, France; D. Theilliol, Centre de Recherche en Automatique de Nancy (CRAN)—CNRS UMR 7039, University of Lorraine, BP 70239, 54506 Vandoeuvre Cedex, France.

I. INTRODUCTION

The objective of control laws is to automatically drive automated systems along desired trajectories without or with reduced human intervention. However, systems in practice have input and/or state constraints and cannot be driven arbitrarily without taking these constraints into consideration. A feasible trajectory is then a trajectory that lies inside the admissible state domain and that does not violate input constraints. If system constraints are not considered, the trajectory may be infeasible and the defined mission may not be accomplished.

Mission unaccomplishment occurs for example when actuators hit their limits and cannot deliver the actuation inputs desired by the controller. Control systems are often designed based on linearized models and do not directly consider amplitude limitations on the control inputs. Then, the presence of input bounds may not only lead to the unaccomplishment of the mission but also can be source of parasitic equilibrium points and limit cycles, or can even lead the closed-loop system to an unstable behavior [1]. Several works considered this problem either by avoiding saturation [1, 2] or allowing saturation while designing stabilizing controllers with saturating controls [3]. Reference management or a reference governor is proposed in the literature for systems with input and/or state related constraints. In [4], a command governor based on conceptual tools of predictive control is designed for solving set-point tracking problems wherein pointwise-in-time input and/or state inequality constraints are present. In [5] a reference governor is designed for general discrete-time and continuous-time nonlinear systems with uncertainties. It relies on safety properties provided by sublevel sets of equilibria-parameterized functions.

In the context of fault-tolerant control (FTC), a reference input management is introduced in [6] to determine appropriate reference inputs in the presence of actuator faults to avoid potential saturation in actuators. The idea is to determine the relationship between the closed-loop control signals u and the associated reference inputs r at steady state and translate the limits of actuator saturation to the desired requirements on the reference inputs. In [7] and [8] an on-line command input adjustment strategy is developed when actuator faults occur. Another reference inputs generation method is proposed in [9]. The reference inputs generation which leads the damaged system to its optimal operating point corresponds to a nonlinear quadratic programming optimization problem. The objective is to minimize the distance between the desirable output vector before and after failure while distributing the most equitably the energy among the healthy actuators.

This work considers the problem of trajectory planning for the quadrotor UAV system. Few research works treat this problem: a time-optimal motion-planning is proposed in [10] to generate a time-optimal motion between two configurations for a nonlinear model of a hovering quadrotor helicopter with four independently driven rotors. In [11] adaptive path planning algorithms are developed for a small unmanned four-rotor helicopter. The path planning for the UAV is processed in different phases. The global preflight planning phase calculates an optimized trajectory in consideration of boundaries. Afterwards, during the flight phase on-board ranging sensors are used to avoid interferences with unknown obstacles. Several trajectory optimization algorithms are presented in [12] for a team of cooperating unmanned vehicles. The algorithms are based on robust receding horizon control and demonstrated in simulation and on two multi-vehicle testbeds using rovers and quadrotors.

Flatness-based trajectory planning/replanning for FTC has been introduced for the first time in the pioneer work of Mai, et al. [13] where dynamic trajectory replanning is employed to avoid systems from hitting their input saturation limits when actuator faults occur. Flatness has been recently employed in trajectory planning for the quadrotor UAV: a method is presented in [14] to generate time-optimal trajectories for the system. In [15] and [16] Cowling, et al. present a quasi-optimal trajectory planner with a simple linear-quadratic regulator (LQR) path following controller. Using the differential flatness, the trajectory planning is posed as a constrained optimization problem in the output space. The key difference among the research works [14], [15], and [16] is the parametrization of the trajectory: in [14] the trajectory is modeled as a composition of a parametric function $P(\lambda)$ defining the path and a monotonically increasing function $\lambda(t)$ specifying the motion on this path. $P(\lambda)$ and $\lambda(t)$ are approximated using B-spline functions and are found using a nonlinear optimization technique. In [15] and [16] several polynomials are investigated such as Laguerre, Chebyshev, and Taylor series expansion polynomials. The choice of the polynomial function to parametrize the trajectory affects the complexity of the optimization problem and the numerical robustness.

In our work the differential flatness is employed to solve the problem of trajectory planning for the quadrotor UAV. The advantage of flatness is that it allows to say a priori if a predefined trajectory is feasible or not and it permits to modify the trajectory parameters in function of the maximal allowable thrusts. The trajectory planning is formulated as an optimization problem with further investigation on trajectory parametrization. For this purpose, the Bézier polynomial function is used to derive algebraic expression for optimization problem constraints. This reduces the complexity of the optimization problem and the calculation requirements and facilitates on-board implementation. The contributions of this work are the following.

1) A minimal-time trajectory planning problem is formulated by using flatness. The objective is to drive the system as fast as possible from an initial position to a final position without hitting the rotors' constraints. The trajectory planning problem is formulated as an optimization problem where algebraic expressions are derived for the constraints thanks to the Bézier polynomial function.
2) The above minimal-time trajectory planning is considered for the fault-free case. When actuator faults occur, FTC is employed to compensate the fault effects [17]. However, a more efficient fault management can be obtained if FTC is combined with trajectory replanning. This paper shows how trajectory replanning can be employed for a more efficient FTC in the presence of actuator faults.
3) A minimal-energy trajectory planning problem is also studied to minimize the energy spent during a given mission.
   The following should be noted.
   a) The trajectory planning problem consists in determining the profile of the reference trajectories to drive the system as fast as possible from an initial position to a final one without hitting actuator constraints. By a "trajectory," we mean the time history of the desired path (the positions and the angle) to be followed by the outputs to be controlled (x, y, z, and the yaw angle ψ). The controls however are not part of the trajectory.
   b) The reference trajectories are chosen as polynomials verifying the boundary conditions and are also function of the mission time t_f. The trajectory planning aims then to tune t_f (and change the profile of the trajectories) so that actuator constraints are not violated. Once t_f is determined, the trajectories output the desired positions and velocities at each time instant.
   c) As shown in the paper, both the controller and the trajectory planning benefit from the flatness property in parameterizing the control inputs in function of the flat outputs. In addition, a flatness-based controller is employed in this work but any other control technique can also be used.

The next section presents the system model. A flatness-based controller and the trajectory planning problem are investigated in Sections III and IV, respectively. Simulation results are given in Section V to verify the feasibility of the proposed approaches and some computing issues are discussed in Section VI.

II. SYSTEM MODEL

This work considers a commonly employed quadrotor UAV model [18]. Note that this is a simplified deterministic model of the more complex and uncertain real system:

$$\begin{aligned}
\ddot{x} &= u_{1x} - \frac{k_1}{m}\dot{x}; & \ddot{\theta} &= u_2 - l\frac{k_4}{J_1}\dot{\theta} \\
\ddot{y} &= u_{1y} - \frac{k_2}{m}\dot{y}; & \ddot{\phi} &= u_3 - l\frac{k_5}{J_2}\dot{\phi} \\
\ddot{z} &= -g + u_1(\cos\phi\cos\theta) - \frac{k_3}{m}\dot{z}; & \ddot{\psi} &= u_4 - \frac{k_6}{J_3}\dot{\psi}
\end{aligned} \tag{1}$$

where $u_{1x} = u_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)$ and $u_{1y} = u_1(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)$. x, y, and z are the coordinates of the quadrotor center of gravity in the Earth-frame. θ, ϕ, and ψ are the pitch, roll, and yaw angles correspondent to the 3-2-1 rotation. m is the mass and J_i $(i = 1,2,3)$ are the moments of inertia along x, y, and z directions. k_i $(i = 1,\ldots,6)$ are the drag coefficients and l is the distance from the center of gravity to each rotor.

u_1 is the linear acceleration applied to the quadrotor in the z-direction of the body frame. u_2, u_3, and u_4 are, respectively, the angular accelerations applied in θ, ϕ, and ψ directions. The accelerations are related to the rotor thrusts as follows:

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} \frac{1}{m} & \frac{1}{m} & \frac{1}{m} & \frac{1}{m} \\ \frac{-l}{J_1} & \frac{-l}{J_1} & \frac{l}{J_1} & \frac{l}{J_1} \\ \frac{-l}{J_2} & \frac{l}{J_2} & \frac{l}{J_2} & \frac{-l}{J_2} \\ \frac{C}{J_3} & \frac{-C}{J_3} & \frac{C}{J_3} & \frac{-C}{J_3} \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{pmatrix} \tag{2}$$

where T_i $(i = 1,\ldots,4)$ is the thrust generated by the ith rotor and C is the thrust-to-moment scaling factor.

A. Model Simplification

Even though the quadrotor model (1) is a simple and deterministic representation of the real system, it remains a relatively complex model to deal with. Nevertheless, several simplified models can be derived to solve the trajectory planning problem and to design a controller that must be robust to model mismatch. First, by assuming small angles θ and ϕ, (1) can be written as

$$\begin{aligned}
\ddot{x} &= u_1(\theta\cos\psi + \phi\sin\psi) - \frac{k_1}{m}\dot{x}; & \ddot{\theta} &= u_2 - l\frac{k_4}{J_1}\dot{\theta} \\
\ddot{y} &= u_1(\theta\sin\psi - \phi\cos\psi) - \frac{k_2}{m}\dot{y}; & \ddot{\phi} &= u_3 - l\frac{k_5}{J_2}\dot{\phi} \\
\ddot{z} &= -g + u_1 - \frac{k_3}{m}\dot{z}; & \ddot{\psi} &= u_4 - \frac{k_6}{J_3}\dot{\psi}.
\end{aligned} \tag{3}$$

Assuming a constant yaw angle ψ, for instance $\psi = 0$:

$$\begin{aligned}
\ddot{x} &= u_1\theta - \frac{k_1}{m}\dot{x}; & \ddot{\theta} &= u_2 - l\frac{k_4}{J_1}\dot{\theta} \\
\ddot{y} &= -u_1\phi - \frac{k_2}{m}\dot{y}; & \ddot{\phi} &= u_3 - l\frac{k_5}{J_2}\dot{\phi} \\
\ddot{z} &= -g + u_1 - \frac{k_3}{m}\dot{z}; & \ddot{\psi} &= u_4 - \frac{k_6}{J_3}\dot{\psi}.
\end{aligned} \tag{4}$$

With negligible drag coefficients at low speeds [10], (4) reads:

$$\begin{aligned}
\ddot{x} &= u_1\theta; & \ddot{\theta} &= u_2 \\
\ddot{y} &= -u_1\phi; & \ddot{\phi} &= u_3 \\
\ddot{z} &= -g + u_1; & \ddot{\psi} &= u_4.
\end{aligned} \tag{5}$$

To decouple the z and x–y axes, one can assume that in hovering condition, $u_1 \approx g$ in the x and y directions [19]. Thus, (5) becomes

$$\begin{aligned}
\ddot{x} &= g\theta; & \ddot{\theta} &= u_2 \\
\ddot{y} &= -g\phi; & \ddot{\phi} &= u_3 \\
\ddot{z} &= -g + u_1; & \ddot{\psi} &= u_4.
\end{aligned} \tag{6}$$

As shown in the subsequent sections, the flatness-based trajectory planning problem is based on determining the maximal thrusts to be applied for a given trajectory. This will not be easy to carry out for the simplified model (4) and thus, calculation load issues will impose to employ the simplified models (5) or (6).

B. Normalization

Before proceeding with the study, the vectors and the matrix of (2) will be normalized, in order to make the problem more understandable and easier to solve. Normalization means that vector components are divided by their maximum values, such that each component is a dimensionless number that lies between 0 and +1 [20]. The first step is to find the maximum values (modules) of the inputs u_i. It can be shown that with $T_{1_{\max}} = T_{2_{\max}} = T_{3_{\max}} = T_{4_{\max}} = T_{\max}$:

$$\begin{aligned}
u_{1_{\max}} &= \frac{4}{m}T_{\max}; & u_{2_{\max}} &= \frac{2l}{J_1}T_{\max} \\
u_{3_{\max}} &= \frac{2l}{J_2}T_{\max}; & u_{4_{\max}} &= \frac{2C}{J_3}T_{\max}.
\end{aligned} \tag{7}$$

The normalized relation is then given by

$$\begin{pmatrix} \frac{u_1}{u_{1_{\max}}} \\ \frac{u_2}{u_{2_{\max}}} \\ \frac{u_3}{u_{3_{\max}}} \\ \frac{u_4}{u_{4_{\max}}} \end{pmatrix} = \begin{pmatrix} +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} \\ -\frac{1}{2} & -\frac{1}{2} & +\frac{1}{2} & +\frac{1}{2} \\ -\frac{1}{2} & +\frac{1}{2} & +\frac{1}{2} & -\frac{1}{2} \\ +\frac{1}{2} & -\frac{1}{2} & +\frac{1}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} \frac{T_1}{T_{\max}} \\ \frac{T_2}{T_{\max}} \\ \frac{T_3}{T_{\max}} \\ \frac{T_4}{T_{\max}} \end{pmatrix}. \tag{8}$$

The normalized form (8) has a number of advantages compared with the standard form (2). First, the components $T_i/T_{\max}$ are dimensionless numbers, restricted to the standard interval $[0,+1]$. This enables better understanding of the problem. Second, all physical parameters are removed from the matrix during the normalization process. This simplifies the calculations.

III. FLATNESS-BASED CONTROL

A flatness-based controller will first be designed for the quadrotor system to track the desired reference trajectories. The flatness property is described as follows [21, 22]. A dynamical system,

$$\dot{x} = f(x,u); \qquad y = h(x) \tag{9}$$

with $x \in \Re^n$ and $u \in \Re^m$, is flat if and only if there exist variables $F \in \Re^m$ called the flat outputs such that $x = \Xi_1(F,\dot{F},\ldots,F^{(n-1)})$, $y = \Xi_2(F,\dot{F},\ldots,F^{(n-1)})$ and $u = \Xi_3(F,\dot{F},\ldots,F^{(n)})$. Ξ_1, Ξ_2, and Ξ_3 are three smooth mappings and $F^{(i)}$ is the ith derivative of F. The parameterization of the control inputs u in function of the flat outputs F plays a key role in the trajectory planning problem: the nominal control inputs to be applied during a mission can be expressed in function of the desired trajectories. Thus allowing to tune the profile of the trajectories for keeping the applied control inputs below the actuator limits.

The determination of the flat outputs is discussed in [23] for single-input/single-output (SISO) and multiple-input/multiple-output (MIMO) linear and nonlinear systems and in [24] for linear systems in polynomial matrix form. In our case it is trivial to see that the outputs to be controlled can be chosen as flat outputs. Thus, the system (3) is flat with flat outputs $F_1 = z$, $F_2 = x$, $F_3 = y$ and $F_4 = \psi$. In addition to x, y, z, and ψ, the parametrization of θ and ϕ in function of the flat outputs is

$$\begin{aligned}
\theta &= \frac{\cos F_4\left(\ddot{F}_2 + \frac{k_1}{m}\dot{F}_2\right) + \sin F_4\left(\ddot{F}_3 + \frac{k_2}{m}\dot{F}_3\right)}{\ddot{F}_1 + \frac{k_3}{m}\dot{F}_1 + g}\\
\phi &= \frac{\sin F_4\left(\ddot{F}_2 + \frac{k_1}{m}\dot{F}_2\right) - \cos F_4\left(\ddot{F}_3 + \frac{k_2}{m}\dot{F}_3\right)}{\ddot{F}_1 + \frac{k_3}{m}\dot{F}_1 + g}.
\end{aligned} \tag{10}$$

The parametrization of the control inputs in function of the flat outputs is

$$\begin{aligned}
u_1 &= \ddot{F}_1 + \frac{k_3}{m}\dot{F}_1 + g; \qquad u_2 = \ddot{\theta} + \frac{lk_4}{J_1}\dot{\theta}\\
u_3 &= \ddot{\phi} + \frac{lk_5}{J_2}\dot{\phi}; \qquad u_4 = \ddot{F}_4 + \frac{k_6}{J_3}\dot{F}_4
\end{aligned} \tag{11}$$

where $\dot{\theta}$, $\ddot{\theta}$, $\dot{\phi}$, and $\ddot{\phi}$ are functions of the flat outputs and can be derived from θ and ϕ given in (10).

A flatness-based control can be derived for the simplified model (3). However, for simplification we assume that the system is not yawing during its mission and thus the simplified model (4) is employed to design the controller. In this case the parametrization of the system states in function of the flat outputs is

$$\begin{aligned}
&x = F_2; \qquad y = F_3; \qquad z = F_1\\
&\theta = \frac{\ddot{F}_2 + \frac{k_1}{m}\dot{F}_2}{\ddot{F}_1 + \frac{k_3}{m}\dot{F}_1 + g}\\
&\phi = -\frac{\ddot{F}_3 + \frac{k_2}{m}\dot{F}_3}{\ddot{F}_1 + \frac{k_3}{m}\dot{F}_1 + g}\\
&\psi = F_4
\end{aligned} \tag{12}$$

and the parametrization of the control inputs in function of the flat outputs is the same as (11), with $\dot{\theta}$, $\ddot{\theta}$, $\dot{\phi}$, and $\ddot{\phi}$ derived from (10).

From the differential parametrization (11), a flatness-based tracking control law can be obtained when replacing $\ddot{F}_1$, $F_2^{(4)}$, $F_3^{(4)}$, and $\ddot{F}_4$ by the auxiliary control inputs $\bar{u}_1$, $\bar{u}_2$, $\bar{u}_3$, and $\bar{u}_4$, respectively, in u_1, u_2, u_3, and u_4, hence:

$$\begin{aligned}
\ddot{F}_1 &= \bar{u}_1; \qquad F_2^{(4)} = \bar{u}_2\\
F_3^{(4)} &= \bar{u}_3; \qquad \ddot{F}_4 = \bar{u}_4
\end{aligned} \tag{13}$$

which renders the system locally linear and decoupled. Let F_1^*, F_2^*, F_3^*, and F_4^* be the reference trajectories for the flat outputs F_1, F_2, F_3, and F_4. Thus, specifying

$$\begin{aligned}
\bar{u}_1 &= \ddot{F}_1^* + K_{11}\dot{e}_1 + K_{12}e_1\\
\bar{u}_2 &= F_2^{(4)*} + K_{21}e_2^{(3)} + K_{22}\ddot{e}_2 + K_{23}\dot{e}_2 + K_{24}e_2\\
\bar{u}_3 &= F_3^{(4)*} + K_{31}e_3^{(3)} + K_{32}\ddot{e}_3 + K_{33}\dot{e}_3 + K_{34}e_3\\
\bar{u}_4 &= \ddot{F}_4^* + K_{41}\dot{e}_4 + K_{42}e_4
\end{aligned} \tag{14}$$

ensures the tracking errors $e_i = F_i^* - F_i$ $(i = 1,\ldots,4)$ to asymptotically converge to zero for an appropriate choice of control gains K_{ij}. The gains K_{ij} can be determined by using pole-placement techniques to ensure good tracking and some robustness to model uncertainties. Finally, the control inputs are

$$\begin{aligned}
u_1 &= \bar{u}_1 + \frac{k_3}{m}\dot{F}_1 + g; \qquad u_2 = \bar{\ddot{\theta}} + \frac{lk_4}{J_1}\dot{\theta}\\
u_3 &= \bar{\ddot{\phi}} + \frac{lk_5}{J_2}\dot{\phi}; \qquad u_4 = \bar{u}_4 + \frac{k_6}{J_3}\dot{F}_4
\end{aligned} \tag{15}$$

where $\bar{\ddot{\theta}}$ is obtained when substituting $F_2^{(4)}$ with $\bar{u}_2$ and $\bar{\ddot{\phi}}$ is obtained when substituting $F_3^{(4)}$ with $\bar{u}_3$. $\bar{u}_i$ $(i = 1,\ldots,4)$ are given in (14). It can be noted that u_i $(i = 1,\ldots,4)$ are function of the derivatives of the flat outputs. These can be derived by using for example an algebraic derivative estimation [25] or sliding mode differentiation [26] when they are not accessible via the measurements.

Several methods can be used to design the reference trajectories F_i^*. In this work we employ the Bézier polynomial function [23]. This choice is motivated by the following two reasons.

1) The coefficients of the polynomial can be easily calculated in function of the initial and the terminal conditions.

2) As shown later, the Bézier polynomial function allows to calculate in a straightforward manner the extrema of the control inputs to be applied along the trajectory.

A general Bézier polynomial function of degree n is

$$F = a_n t^n + a_{n-1}t^{n-1} + \cdots + a_2 t^2 + a_1 t + a_0 \qquad (16)$$

where t is the time and a_i $(i = 0,\ldots,n)$ are constant coefficients to be calculated in function of the initial and final conditions. It is clear that the larger n is, the smoother the reference trajectory is. However, calculations for trajectory planning become heavier as n increases.

For the quadrotor UAV, it can be seen in (11) that u_1 is function of $\ddot{F}_1$, u_2 of $F_2^{(4)}$, u_3 of $F_3^{(4)}$, and u_4 of $\ddot{F}_4$. The relative degrees are then $r_1 = 2$, $r_2 = 4$, $r_3 = 4$, and $r_4 = 2$. Smooth control inputs u_i can be obtained if one can impose $F_i^{(0)}$ up to $F_i^{(n_i)}$ at the initial and final time (i.e., up to $F_i^{(n_i)}(t_0)$ and $F_i^{(n_i)}(t_f)$) where $n_i = 2$ for $i = 1,4$ and $n_i = 4$ for $i = 2,3$. To this end, we employ a Bézier polynomial function of degree 5 for F_1 and F_4 and a Bézier polynomial function of degree 9 for F_2 and F_3. The reference trajectories are then:

$$F_i^* = a_5^i t^5 + a_4^i t^4 + a_3^i t^3 + a_2^i t^2 + a_1^i t + a_0^i; \qquad (i = 1,4) \qquad (17)$$

and

$$F_i^* = a_9^i t^9 + a_8^i t^8 + a_7^i t^7 + \cdots + a_1^i t + a_0^i; \qquad (i = 2,3). \qquad (18)$$

The coefficients a_j^i $(i = 1,4$ and $j = 0,\ldots,5)$ are calculated to verify the initial conditions $F_i(t_0)$, $\dot{F}_i(t_0)$, $\ddot{F}_i(t_0)$ and the final conditions $F_i(t_f)$, $\dot{F}_i(t_f)$, $\ddot{F}_i(t_f)$. The coefficients a_j^i $(i = 2,3$ and $j = 0,\ldots,9)$ are calculated to verify the initial conditions $F_i(t_0)$, $\dot{F}_i(t_0)$, $\ddot{F}_i(t_0)$, $F_i^{(3)}(t_0)$, $F_i^{(4)}(t_0)$ and the final conditions $F_i(t_f)$, $\dot{F}_i(t_f)$, $\ddot{F}_i(t_f)$, $F_i^{(3)}(t_f)$, $F_i^{(4)}(t_f)$. t_0 and t_f are, respectively, the initial and the final instants of the mission.

IV. FLATNESS-BASED TRAJECTORY PLANNING/REPLANNING

The basic idea of flatness-based trajectory planning is to parametrize the control inputs to be applied in function of the reference trajectory to be followed. This allows to tune the parameters of the desired trajectory so that the actuator constraints are not violated. This section investigates the trajectory planning problem for both cases: fault-free case and fault case. In the fault-free case, the objective is to drive the system from an initial position to a final one while minimizing the total time of the mission or the energy spent during the mission. When actuator faults occur, the reference trajectory is replanned to take into consideration the remaining resources in the system after the fault for allowing a better fault management.

A. Fault-Free Case: Minimal-Time Trajectory Planning

The flatness-based trajectory planning by using (11) and (10) appears to be a relatively complex problem to be solved. Thus, the simplified model (5) is employed by assuming small drag terms. In this case the nominal control inputs u_i^* to be applied for the nominal reference trajectories F_i^* are given by

$$\begin{aligned} u_1^* &= \ddot{F}_1^* + g; \qquad & u_2^* &= \ddot{\theta}^* \\ u_3^* &= \ddot{\phi}^*; \qquad & u_4^* &= \ddot{F}_4^* \end{aligned} \qquad (19)$$

where

$$\ddot{\theta}^* = \frac{F_2^{(4)*}}{\ddot{F}_1^* + g} - 2\frac{F_2^{(3)*}F_1^{(3)*}}{(\ddot{F}_1^* + g)^2} + 2\frac{\ddot{F}_2^*(F_1^{(3)*})^2}{(\ddot{F}_1^* + g)^3} - \frac{\ddot{F}_2^* F_1^{(4)*}}{(\ddot{F}_1^* + g)^2} \qquad (20)$$

$$\ddot{\phi}^* = -\frac{F_3^{(4)*}}{\ddot{F}_1^* + g} + 2\frac{F_3^{(3)*}F_1^{(3)*}}{(\ddot{F}_1^* + g)^2} - 2\frac{\ddot{F}_3^*(F_1^{(3)*})^2}{(\ddot{F}_1^* + g)^3} + \frac{\ddot{F}_3^* F_1^{(4)*}}{(\ddot{F}_1^* + g)^2}. \qquad (21)$$

It can be seen that in steady state (F_i^*=constant), the control inputs to be applied are

$$\begin{aligned} u_1^* &= g; \qquad & u_2^* &= 0 \\ u_3^* &= 0; \qquad & u_4^* &= 0. \end{aligned} \qquad (22)$$

Let us denote $\underline{T}_i = T_i/T_{\max}$. Taking the inverse of (8), one can note that the nominal $\underline{T}_i^*$ to be applied along the reference trajectories F_i^* are

$$\begin{aligned} \underline{T}_1^* &= \frac{u_1^*}{u_{1_{\max}}} - \frac{1}{2}\frac{u_2^*}{u_{2_{\max}}} - \frac{1}{2}\frac{u_3^*}{u_{3_{\max}}} + \frac{1}{2}\frac{u_4^*}{u_{4_{\max}}} \\ \underline{T}_2^* &= \frac{u_1^*}{u_{1_{\max}}} - \frac{1}{2}\frac{u_2^*}{u_{2_{\max}}} + \frac{1}{2}\frac{u_3^*}{u_{3_{\max}}} - \frac{1}{2}\frac{u_4^*}{u_{4_{\max}}} \\ \underline{T}_3^* &= \frac{u_1^*}{u_{1_{\max}}} + \frac{1}{2}\frac{u_2^*}{u_{2_{\max}}} + \frac{1}{2}\frac{u_3^*}{u_{3_{\max}}} + \frac{1}{2}\frac{u_4^*}{u_{4_{\max}}} \\ \underline{T}_4^* &= \frac{u_1^*}{u_{1_{\max}}} + \frac{1}{2}\frac{u_2^*}{u_{2_{\max}}} - \frac{1}{2}\frac{u_3^*}{u_{3_{\max}}} - \frac{1}{2}\frac{u_4^*}{u_{4_{\max}}} \end{aligned} \qquad (23)$$

where $u_{i_{\max}}$ are constants and are given in (7) and u_i^* are the nominal control inputs and are given in (19).

Once the nominal $\underline{T}_i^*$ in (23) are expressed in function of the reference trajectories F_i^*,the profile of F_i^* can be tuned to drive the system as fast as possible from an initial position to a final one without hitting actuator constraints. This profile depends on the coefficients a_j^i of the reference trajectories which are defined as polynomial functions of fixed degrees in (22) and (23). The coefficients are in function of the initial and final conditions as well as t_0 and t_f where the final time t_f is the sole unknown parameter and thus, they can be obtained by determining t_f. The trajectory planning problem consists then in determining the minimal travel time to drive the system from an initial position to a final position without violating the constraints of the UAV rotors. Mathematically, this can be expressed as an optimization problem where the objective function to minimize is $t_f - t_0$, the decision variable is t_f, and the constraints are the actuator limitations:

$$P\begin{cases}\text{Minimize} & t_f - t_0\\ \text{Subject to} & 0 \le \underline{T}_i^* \le 1; \qquad i = (1,\ldots,4)\end{cases} \tag{24}$$

where $\underline{T}_i^*$ is the ith normalized thrust with a maximal allowable value equals to 1 after normalization. The nominal thrust $\underline{T}_i^*$ is function of time t. Thus, solving problem (24) requires using calculus of variations which may impose heavy calculations. As an alternative solution, we propose replacing $\underline{T}_i^*$ of (24) by their extrema where the extrema collectively denote the minima and the maxima of $\underline{T}_i^*(t)$ and to solve (25) instead of (24):

$$P\begin{cases}\text{Minimize} & t_f - t_0\\ \text{Subject to} & 0 \le \underline{T}_{i_{\text{Ext}}}^* \le 1; \qquad i = (1,\ldots,4)\\ & 0 \le \underline{T}_i^*(t_0) \le 1\\ & 0 \le \underline{T}_i^*(t_f) \le 1\end{cases} \tag{25}$$

where $\underline{T}_{i_{\text{Ext}}}^*$ is the extrema of $\underline{T}_i^*$ and $\underline{T}_i^*(t_0)$ and $\underline{T}_i^*(t_f)$ represent the thrusts applied at t_0 and t_f, respectively. The extrema $\underline{T}_{i_{\text{Ext}}}^*$ are algebraic expressions that are in function of t_0, t_f, the initial and final conditions. Thus, using (25) instead of (24) for trajectory planning might be more adequate for real time on board implementation: when a planning is required, it is sufficient to plug the parameters (t_0, the initial and final conditions) with their numerical values in (25) to get the solution t_f. Hence, reducing calculation requirements and making the trajectory planner more suitable for implementation on the quadrotor's on-board microcontroller.

The extrema can be obtained as follows $\underline{T}_{i_{\text{Ext}}}^* = \underline{T}_i^*(z_i^*)$ where z_i^* are the critical points and are the solutions of

$$\frac{d\underline{T}_i^*(t)}{dt} = 0. \tag{26}$$

If $\underline{T}_{i_{\text{MIN}}}^*$ and $\underline{T}_{i_{\text{MAX}}}^*$ are defined as the global minima and the global maxim,a respectively, then one can say that $\underline{T}_{i_{\text{MIN}}}^* \le \underline{T}_i^*(t) \le \underline{T}_{i_{\text{MAX}}}^*$ $\forall t \in [t_0, t_f]$ and thus, the solution of the problem (25) is also a solution of (24). It can be noted here that $\underline{T}_i^*(t)$ is a polynomial equation and that the degree of $d\underline{T}_i^*(t)/dt$ is not less than five. According to the theorem of Abel and Ruffini [27]:

THEOREM 1 *A general algebraic equation of degree* ≥ 5 *cannot be solved in radicals. This means that there does not exist any formula which would express the roots of such equation as functions of the coefficients by means of the algebraic operations and roots of natural degrees.*

This means that in general, polynomial equations higher than fourth degree are incapable of algebraic solution in terms of a finite number of additions, subtractions, multiplications, divisions, and root extractions [28]. Thus, the algebraic extrema $\underline{T}_{i_{\text{Ext}}}^*$ for $\underline{T}_i^*(t)$ cannot be derived by using model (5). Let us consider the simplified model (6) by assuming a hovering condition. In this case the parametrization of the control inputs in function of the flat outputs is

$$\begin{aligned} u_1 &= \ddot{F}_1 + g; \qquad u_2 = \frac{1}{g}F_2^{(4)}\\ u_3 &= -\frac{1}{g}F_3^{(4)}; \qquad u_4 = \ddot{F}_4. \end{aligned} \tag{27}$$

From the differential parametrization (27), it can be concluded that the nominal control inputs u_i^* to be applied for the nominal reference trajectories F_i^* are given by

$$\begin{aligned} u_1^* &= \ddot{F}_1^* + g; \qquad u_2^* = \frac{1}{g}F_2^{(4)*}\\ u_3^* &= -\frac{1}{g}F_3^{(4)*}; \qquad u_4^* = \ddot{F}_4^*. \end{aligned} \tag{28}$$

With (28), $\underline{T}_i^*(t)$ is a polynomial equation of degree five and $d\underline{T}_i^*(t)/dt$ is of degree four. Thus, an algebraic $\underline{T}_{i_{\text{Ext}}}^*$ can be derived to solve the optimization problem (25).

The above study is carried out using the nominal control input (28). However, in practice, an additional control term is used to tackle uncertainties and to ensure the stability of the closed-loop system (see (14)). To take into consideration the control input generated by the additional control term, a safety margin is created by introducing a constant ρ in the optimization problem as follows:

$$P\begin{cases}\text{Minimize} & t_f - t_0\\ \text{Subject to} & \rho \le \underline{T}_{i_{\text{Ext}}}^* \le 1-\rho; \qquad i = (1,\ldots,4)\\ & \rho \le \underline{T}_i^*(t_0) \le 1-\rho\\ & \rho \le \underline{T}_i^*(t_f) \le 1-\rho\end{cases} \tag{29}$$

where $0 \le \rho < 1$ depends on the model uncertainties. It is clear that the choice of ρ requires to quantify model uncertainties which is not an easy task. In problem (29) the extrema $\underline{T}_{i_{\text{Ext}}}^*$ are derived by using the simplified model (6) which is obtained by assuming small pitch and roll angles. Thus, the choice of ρ can be made easier if one can reduce model uncertainties by forcing the system to verify this assumption (small pitch and roll angles) during the mission. Thanks to the system's flatness, the angles θ and ϕ can be parametrized in function of the flat outputs (see (6)). Thus, one can obtain the nominal angles in function of the reference trajectories:

$$\theta^* = \frac{1}{g}\ddot{F}_2^*; \qquad \phi^* = -\frac{1}{g}\ddot{F}_3^*. \tag{30}$$

The objective is then to ensure that $-\bar{\theta} \leq \theta^*(t) \leq +\bar{\theta}$ and $-\bar{\phi} \leq \phi^*(t) \leq +\bar{\phi}$ $\forall t \in [t_0,t_f]$ where $\bar{\theta} > 0$ and $\bar{\phi} > 0$ are constants to be imposed. Similarly as before, define θ^*_{Ext} and ϕ^*_{Ext} as the extrema of $\theta^*(t)$ and $\phi^*(t)$ where $\theta^*_{\text{Ext}} = \theta^*(z^*_\theta)$, $\phi^*_{\text{Ext}} = \phi^*(z^*_\phi)$, and z^*_θ and z^*_ϕ are the solutions of

$$\frac{d\theta^*(t)}{dt} = 0; \qquad \frac{d\phi^*(t)}{dt} = 0. \tag{31}$$

With two additional constraints, the optimization problem becomes

$$P\begin{cases} \text{Minimize} & t_f - t_0 \\ \text{Subject to} & \rho \leq \underline{T}^*_{i_{\text{Ext}}} \leq 1-\rho; \qquad i = (1,\ldots,4) \\ & \rho \leq \underline{T}^*_i(t_0) \leq 1-\rho \\ & \rho \leq \underline{T}^*_i(t_f) \leq 1-\rho \\ & -\bar{\theta} \leq \theta^*_{\text{Ext}} \leq +\bar{\theta} \\ & -\bar{\phi} \leq \phi^*_{\text{Ext}} \leq +\bar{\phi} \end{cases}. \tag{32}$$

It should be noted that solving the optimization problem for trajectory planning is performed only once before starting the mission. The trajectory planning using the simplified model (6) can be performed either at the ground station or at the quadrotor's on-board microcontroller. In the former case, the trajectory parameters as well as the optimal time mission t_f solution of the problem are communicated to the UAV via a wireless communication.

B. Fault-Free Case: Minimal-Energy Trajectory Planning

The section above considers the problem of trajectory planning with the objective to minimize the final time of the mission t_f. However, due to the limited payload of the UAV, the on-board batteries are small and have limited capacity. Thus, energy saving is of great importance to increase the system's autonomy. This section investigates then the problem of minimal-energy trajectory planning for the quadrotor UAV.

It is not an easy task to derive a formal equation for the energy spent during a mission and if such an equation exists, it would be nonlinear and not easy to deal with. However, it is obvious that the thrust and the consumed energy are somewhat proportional, i.e., the more thrust is generated, the more energy is consumed and vice-versa. Thus, thrust can be considered as an "image" of the consumed energy and can be directly employed for the minimal-energy analysis. The minimal-energy planning consists then in minimizing the index:

$$J = \int_{t_0}^{t_f} \underline{T}^T Q \underline{T}\, dt \tag{33}$$

where $\underline{T}^T = [\underline{T}_1\ \underline{T}_2\ \underline{T}_3\ \underline{T}_4]$ and Q is a weighting matrix. For simplicity Q is chosen as I_4, thus (33) can be written as

$$J = \int_{t_0}^{t_f} \sum_{i=1}^{4} \underline{T}_i^2 dt. \tag{34}$$

With $\underline{T}_i = T_i/T_{\max}$ given in the inverse of (8), it is easy to see that:

$$\sum_{i=1}^{4} \underline{T}_i^2 = 4\underline{u}_1^2 + \underline{u}_2^2 + \underline{u}_3^2 + \underline{u}_4^2 \tag{35}$$

where $\underline{u}_i = u_i/u_{i_{\max}}$ $(i = 1,\ldots,4)$. For the nominal desired trajectories F_i^*,the energy to be spent is

$$\sum_{i=1}^{4} \underline{T}_i^{*2} = 4\underline{u}_1^{*2} + \underline{u}_2^{*2} + \underline{u}_3^{*2} + \underline{u}_4^{*2} \tag{36}$$

where $\underline{u}_i^* = u_i^*/u_{i_{\max}}$ and u_i^* are given in (28). Equation (36) is a polynomial function of t, t_0, t_f, and initial and final conditions. Thus, it is easy to say that:

$$J = E(t)|_{t_0}^{t_f} = E(t_f) - E(t_0) \tag{37}$$

where $E(t)$ is the primitive:

$$E(t) = \int (4\underline{u}_1^{*2} + \underline{u}_2^{*2} + \underline{u}_3^{*2} + \underline{u}_4^{*2}) dt. \tag{38}$$

Finally, the minimal-energy trajectory planning consists in solving the optimization problem:

$$P\begin{cases} \text{Minimize} & E(t_f) - E(t_0) \\ \text{Subject to} & \rho \leq \underline{T}^*_{i_{\text{Ext}}} \leq 1-\rho; \qquad i = (1,\ldots,4) \\ & \rho \leq \underline{T}^*_i(t_0) \leq 1-\rho \\ & \rho \leq \underline{T}^*_i(t_f) \leq 1-\rho \end{cases}. \tag{39}$$

A multi-objective optimization problem can also be used to combine both energy and total time of the mission as follows:

$$P\begin{cases} \text{Min} & \alpha(t_f - t_0) + (1-\alpha)(E(t_f) - E(t_0)) \\ \text{s.t.} & \rho \leq \underline{T}^*_{i_{\text{Ext}}} \leq 1-\rho; \qquad i = (1,\ldots,4) \\ & \rho \leq \underline{T}^*_i(t_0) \leq 1-\rho \\ & \rho \leq \underline{T}^*_i(t_f) \leq 1-\rho \end{cases} \tag{40}$$

where $\alpha \in [0,1]$ allows a compromise between energy and time.

In our case, the total energy to be spent during the mission can be divided into two parts: the first part serves in counteracting the gravity and keeping the system in the xy-plane. The second part serves in tracking the desired trajectory and driving the system to the final desired position. The first part is proportional to the time of the mission, i.e., it increases if t_f increases and vice-versa. Thus, one expects the optimization problem to minimize t_f as much as possible. On the other hand, a very small t_f will result in 1) large accelerations and velocities and in consequence large amplitude for the second part of the energy and 2) large thrusts that may hit the rotors constraints. Thus, the solution of problem (39) is a compromise between the two parts. This point is further clarified in the simulation section.

C. Fault Case: Minimal-Time Trajectory Replanning

This section investigates the trajectory replanning problem when faults occur. The trajectory replanning aims to redefine the trajectory of the system during the mission in function of the remaining resources. The objective is to provide a feasible trajectory for the UAV to follow while avoiding actuator saturation. The approach considers the minimal-time replanning but the energy can be easily considered as shown in the previous section. The actuator faults considered in this work are the loss of effectiveness. Such faults can be modeled as

$$\underline{T}_i^f = (1 - \delta_i)\underline{T}_i \tag{41}$$

where $\delta_i \in [0, 1]$. For $\delta_i = 0$, the ith rotor is healthy and for $\delta_i = 1$, the ith rotor is totally lost. According to the definition of $\underline{T}_i$ ($0 \leq \underline{T}_i \leq 1$), we have

$$0 \leq (1 - \delta_i)\underline{T}_i \leq (1 - \delta_i). \tag{42}$$

It is important to determine the interval of δ_i for which the stability of the quadrotor can be maintained. It is shown in the previous section that the control inputs to be applied in steady state ($F_i^* =$ constant) are

$$\begin{aligned} u_1^* &= g; \qquad u_2^* = 0 \\ u_3^* &= 0; \qquad u_4^* = 0. \end{aligned} \tag{43}$$

Thus, according to (23), the nominal thrust $\underline{T}_i^*$ ($i = 1, \ldots, 4$) to be applied in steady state is

$$\underline{T}_i^* = \frac{u_1^*}{u_{1_{\max}}} = \frac{g}{u_{1_{\max}}}. \tag{44}$$

To maintain the system height when a fault occurs, one should have

$$(1 - \delta_i) \geq \frac{g}{u_{1_{\max}}}. \tag{45}$$

Substituting $u_{1_{\max}}$ by $4T_{\max}/m$, one obtains that the system height can be maintained if the loss of effectiveness:

$$\delta_i \leq 1 - \frac{mg}{4T_{\max}} < 1 \tag{46}$$

where the mass m and the maximum thrust $T_{\max}$ are known and constant. To maximize the upper bound $1 - (mg)/(4T_{\max})$ and allow a bigger δ_i, one can minimize the mass m or maximize the maximum thrust $T_{\max}$. If for some fault, (46) is not verified, then the system height cannot be maintained.

Finally, the trajectory replanning in the fault case can be performed by taking into consideration the new constraints of the rotors, i.e., solving the optimization problem:

$$P \begin{cases} \text{Min} & t_f - t_0 \\ \text{s.t.} & \rho \leq \underline{T}^*_{i_{\text{Ext}}} \leq 1 - \delta_i - \rho; \qquad i = (1, \ldots, 4) \\ & \rho \leq \underline{T}_i^*(t_0) \leq 1 - \delta_i - \rho \\ & \rho \leq \underline{T}_i^*(t_f) \leq 1 - \delta_i - \rho \\ & -\bar{\theta} \leq \theta^*_{\text{Ext}} \leq +\bar{\theta} \\ & -\bar{\phi} \leq \phi^*_{\text{Ext}} \leq +\bar{\phi} \end{cases}. \tag{47}$$

The above problem is to be solved on-line after the detection, isolation, and identification of the fault. Due to the rapidity and the instability of the system, it is important to perform trajectory replanning on-board and not at the ground station.

It should be noted here that the baseline controller may be able to deal with minor actuator faults (such as minor loss of effectiveness) based on the inherent robustness of the controller. In this case, an FTC will not be needed. However, for more critical faults (jammed rotor, major or complete loss of effectiveness), the controller may fail to maintain the stability of the system and thus an FTC is necessary. On the other hand, trajectory replanning cannot replace the FTC and combining both of them offers a better fault management by changing the reference trajectory in function of the available resources.

In this work we assume that an FDD (fault detection and diagnosis) module is present to detect, isolate, and identify the amplitude of the fault. Nevertheless, one can see the following.

1) Some FTC approaches require an information about the occurred fault. Thus, an FDD module must be present to provide the information about the location of the fault, the type of the fault, and its amplitude. Other approaches do not require an explicit information about the fault and in this case, the perfect knowledge of the degradation of the rotor is not required.

2) Solving the optimization problem (47) requires the knowledge of $\delta_i + \rho$. Here, actuator faults can be viewed as uncertainties in the model parameters and thus trajectory replanning can be interpreted as seeking an upper bound for $\delta_i + \rho$. In this case the perfect knowledge of the degradation of the rotor δ_i gives the best solution (the smallest possible t_f) but

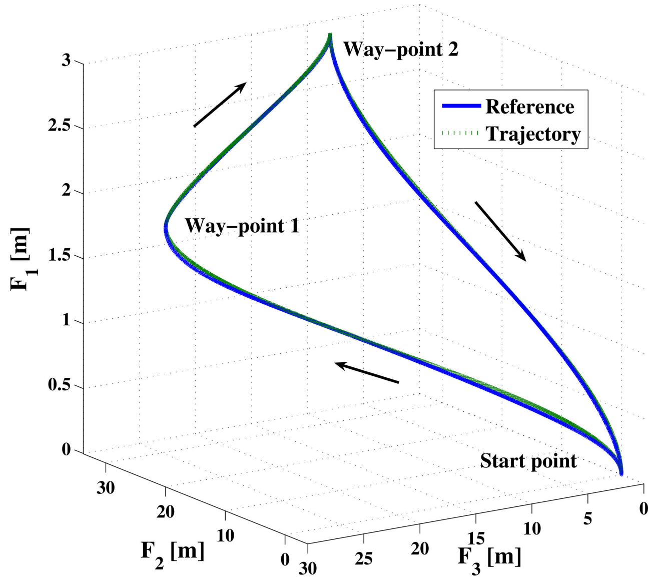

Fig. 1. Three-directional trajectory of quadrotor UAV.

this knowledge is not of extreme importance since any $\Delta \geq \delta_i$ will give an acceptable solution (in this case, t_f will be larger but in the fault case the safety of the system/mission comes first).

V. SIMULATION RESULTS

This section presents different simulation results to show the effectiveness of the proposed approaches. The simulation is performed using MATLAB/Simulink with solver ODE1 and a fixed step size of 0.01 s. Band-limited white noise is added to simulate sensor noise. The numerical values of the parameters are derived from [18] with $T_{\max}$ assumed to be equal to $8N$.

It should be noted here that the flatness-based controller and trajectory planning approaches are developed using simplified models of the quadrotor UAVs. However, the approaches are applied to the nonlinear model (1) unless otherwise stated. This allows introducing some model uncertainties and testing the robustness of the approaches.

A. Flatness-Based Control

This simulation tests for the flatness-based controller developed in Section III. The quadrotor is required to take-off from an initial position, visit two way-points and then land back to the initial position. It should be noted here that at the way-points we have $\dot{F}_i(t_0) = \ddot{F}_i(t_0) = 0$ $(i = 1,4)$ and $\dot{F}_i(t_0) = \ddot{F}_i(t_0) = F_i^{(3)}(t_0) = F_i^{(4)}(t_0) = 0$ $(i = 2,3)$. Figure 1 shows the three-directional trajectory of the system.

Figure 2 shows the applied thrusts which are obtained with a Bézier polynomial function of degree 5 for F_1^* and of degree 9 for F_2^* and F_3^*. As discussed in Section III, the obtained thrusts are smooth due to the choice of smooth desired trajectories for the flat outputs.

The results above show the effectiveness of the controller. The trajectory is determined by the Bézier polynomial function but the time t_f of the mission is not optimal. The next section investigates the minimal-time trajectory planning.

B. Minimal-Time Three-Directional Trajectory Planning

The three-directional trajectory planning is performed using the simplified model (6) and the optimization problem (29). The objective is to drive the system from an initial position to a final one in a minimal time while not violating actuator constraints. The initial conditions for the desired trajectories are $F_1(t_0) = \dot{F}_1(t_0) = \ddot{F}_1(t_0) = F_2(t_0) = \dot{F}_2(t_0) = \ddot{F}_2(t_0) = F_2^{(3)}(t_0) = F_2^{(4)}(t_0) = F_3(t_0) = \dot{F}_3(t_0) = \ddot{F}_3(t_0) = F_3^{(3)}(t_0) = F_3^{(4)}(t_0) = F_4(t_0) = \dot{F}_4(t_0) = \ddot{F}_4(t_0) = 0$. The final conditions are $F_1(t_f) = 4$ m, $F_2(t_f) = 20$ m, $F_3(t_f) = 30$ m and $\dot{F}_1(t_f) = \ddot{F}_1(t_f) = \dot{F}_2(t_f) = \ddot{F}_2(t_f) = F_2^{(3)}(t_f) = F_2^{(4)}(t_f) = \dot{F}_3(t_f) = \ddot{F}_3(t_f) = F_3^{(3)}(t_f) = F_3^{(4)}(t_f) = F_4(t_f) = \dot{F}_4(t_f) = \ddot{F}_4(t_f) = 0$.

Since the simplified model (6) is employed in the three-directional trajectory planning, we choose to set the parameter ρ to 0.2 to take into consideration the mismatch between (6) and the nonlinear model (1). In this case, the obtained solution is $t_f = 7.39$ s. Figure 3 shows the three-directional trajectory of the system and Fig. 4 shows the applied thrusts along the trajectory. As expected, the applied thrusts exceed the safety margin $1-\rho = 0.8$ (due to model mismatch) but remain less than 1 (i.e., they do not hit the actuator constraints).

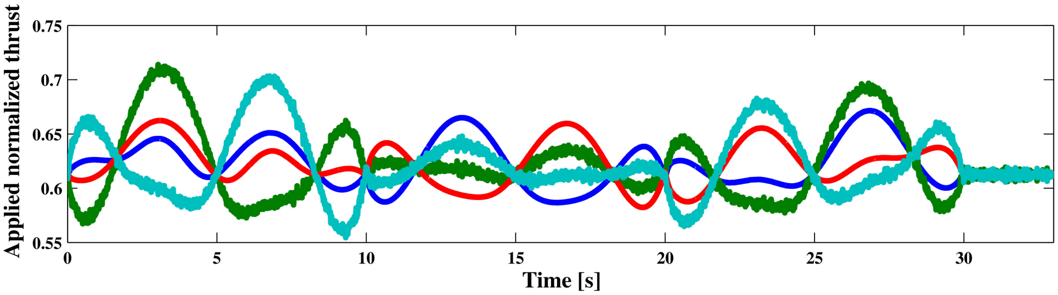

Fig. 2. Applied normalized thrusts $\underline{T}_i$ for smoother trajectories.

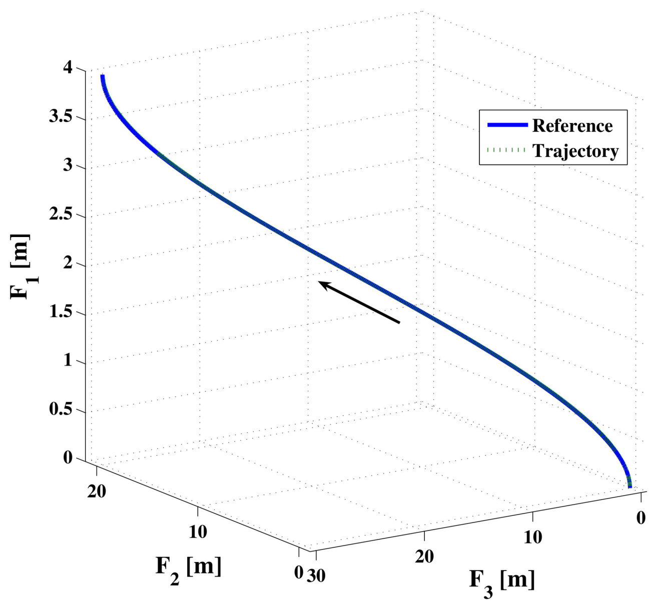

Fig. 3. Three-directional trajectory of the quadrotor.

C. Minimal-Energy Three-Directional Trajectory Planning

The minimal-energy trajectory planning problem aims to find the time t_f of the mission so that the total energy spent during the mission is minimized. This is illustrated here by using the approach presented in Section IV-B.

The approach is first tested using the optimization problem (39) without considering the constraints. The trajectory parameters are the same as in Section V-B. The solution obtained is t_f = 5.96 s and the corresponding value of the index J is J = 10.24. As expected, the optimization problem tries to minimize t_f as much as possible to reduce the energy spent on counteracting the system weight. However, a very small t_f will result in large accelerations and velocities (when driving the system between two way-points) and lead in consequence to large energy consumption. Thus, the solution t_f = 5.96 s is a compromise between two aspects: counteracting the gravity and tracking the desired trajectory. However, the solution t_f = 5.96 s is smaller than t_f = 7.39 s obtained in the fault-free trajectory planning (Section V-B). Thus, it can be noted that actuator constraints will be hit if the solution of the unconstrained minimal-energy trajectory planning problem is employed.

In a second step, the constraints are considered in the optimization problem (39). The solution obtained is t_f = 7.39 s and the corresponding value of the index J is J = 11.38. It can be noted that the solution t_f = 7.39 s is the same as the solution of the minimal-time trajectory planning problem. This solution has a worse index J but it is the best solution (the least t_f) that can be obtained in the presence of actuator constraints.

As a consequence of the results given above, one can conclude that the solution obtained from the minimal-time trajectory planning is also the best solution to minimize the energy spent during a mission.

D. Minimal-Time Trajectory Planning in xy-Plane: Fault-Free Case

The simulation is done with unconstrained angles θ and ϕ using the optimization problem (29). The initial conditions for the desired trajectories are $F_1(t_0) \neq 0$ and $\dot{F}_1(t_0) = \ddot{F}_1(t_0) = F_2(t_0) = \dot{F}_2(t_0) = \ddot{F}_2(t_0) = F_2^{(3)}(t_0) = F_2^{(4)}(t_0) = F_3(t_0) = \dot{F}_3(t_0) = \ddot{F}_3(t_0) = F_3^{(3)}(t_0) = F_3^{(4)}(t_0) = F_4(t_0) = \dot{F}_4(t_0) = \ddot{F}_4(t_0) = 0$. The final conditions are $F_1(t_f) = F_1(t_0)$, $F_2(t_f)$ = 20 m, $F_3(t_f)$ = 30 m and $\dot{F}_1(t_f) = \ddot{F}_1(t_f) = \dot{F}_2(t_f) = \ddot{F}_2(t_f) = F_2^{(3)}(t_f) = F_2^{(4)}(t_f) = \dot{F}_3(t_f) = \ddot{F}_3(t_f) = F_3^{(3)}(t_f) = F_3^{(4)}(t_f) = F_4(t_f) = \dot{F}_4(t_f) = \ddot{F}_4(t_f) = 0$. The parameter ρ is set to 0.1 and the obtained solution is t_f = 6.81 s.

The approach was first applied to the linear model (6). Figure 5 illustrates the x and y positions of the quadrotor during the mission. The flatness-based controller is able to drive the system from the initial configuration to the final desired one. Figure 6 shows the applied thrusts where it can be noted that $\rho \leq \underline{T}_i \leq 1 - \rho \; \forall t \in [t_0, t_f]$. It can be seen that after the arrival of the system to its final configuration, non-zero thrusts are applied to keep the system in hovering.

In a second step the simulation is performed with the nonlinear model (1). The evolution of the x and y positions are the same as in the linear case (Fig. 5). The applied thrusts are shown in Fig. 7. It can be noted that the thrusts are slightly different than the linear case but $\underline{T}_i$ ($i = 1,\ldots,4$) remain in the admissible domain (i.e., $0 \leq \underline{T}_i \leq 1 \; \forall t \in [t_0, t_f]$) regardless of the model uncertainties.

Model uncertainties have two sources: the nonsmall angles θ and ϕ and the drag terms. To deal with model uncertainties, two solutions are

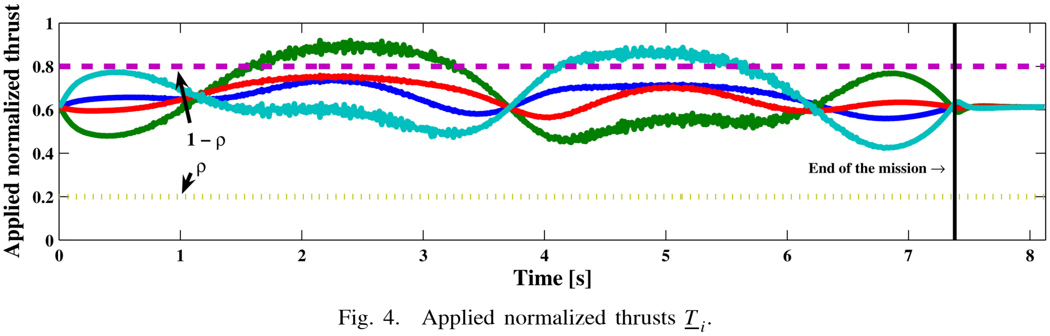

Fig. 4. Applied normalized thrusts $\underline{T}_i$.

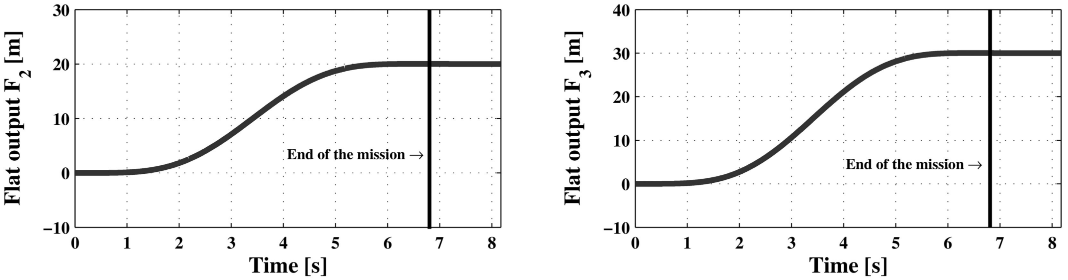

Fig. 5. Flat output F_2 and F_3 of linear model.

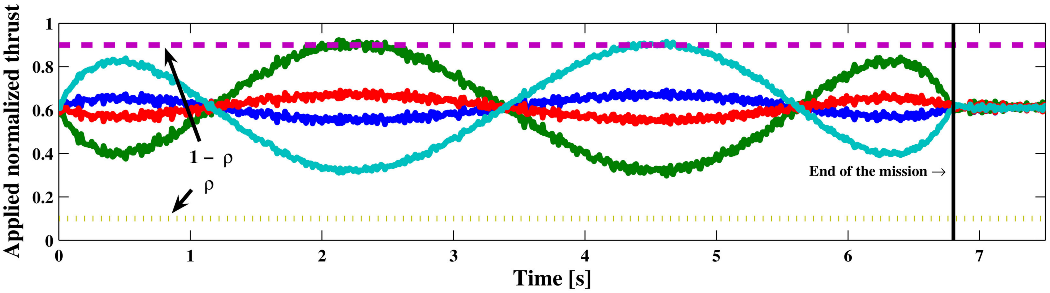

Fig. 6. Applied normalized thrusts $\underline{T}_i$ of linear model.

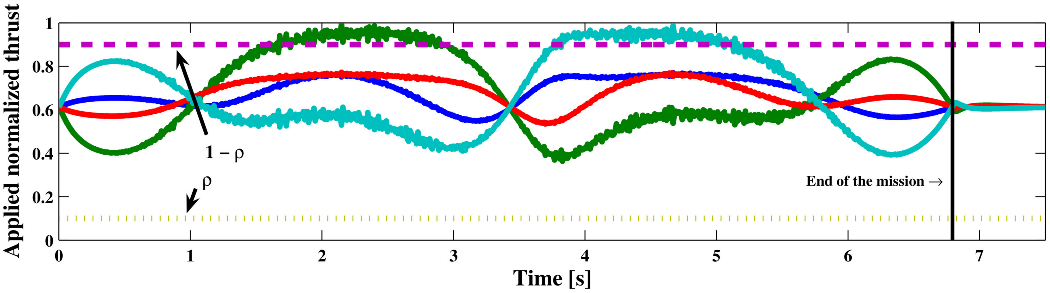

Fig. 7. Applied normalized thrusts $\underline{T}_i$ of the nonlinear model.

possible. The first solution is to increase ρ. This will increase the safety margin and in consequence the time t_f of the mission. A larger t_f will result in reduced velocity of the system, reduced variation of the angles θ and ϕ, and reduced drag terms (then reduced uncertainties). The second solution consists in employing the optimization problem (32) to impose an admissible domain for the angles θ and ϕ. A similar study can be carried out to impose small velocities and angular rates to minimize the contribution of the drag terms (see (1)). This need to deal with model uncertainties rises up when applying the approach to the real system.

E. Minimal-Time Trajectory Planning in xy-Plane: Fault Case

The trajectory planning (or replanning) in the fault case consists in solving the optimization problem (47) where the initial conditions are those of the replanning instant and the final conditions are the prefault one. Note that the final conditions under the prefault case can be attained if sufficient resources are available after the occurrence of a fault.

As previously explained, the system height can be maintained if (46) is verified. With a mass of 2 kg (see [18]), it can be seen that the maximal allowable loss of effectiveness is $\delta_{i_{\max}} = 0.3875$. However, it should be noted that the optimization problem (47) is infeasible if $\delta_i + \rho > 0.3875$ because no solution can be found to stabilize the system in steady state.

The trajectory replanning is first applied to the linear model (6). The fault-free evolution of the system is given in Figs. 5 and 6. For the fault case, the fourth actuator is assumed to lose 25% ($\delta_4 = 0.25$) of its effectiveness at time instant $t = 2$ s. Figure 8 shows the system behavior with the fault and Fig. 9 shows the corresponding thrusts. In the presence of an actuator fault, the controller tries to follow the reference trajectory but the generated thrusts hit the rotor constraints which are more severe due to the fault. Thus, the system is not able to reach the desired final position ($F_2^*(t_f) = 20$ m and $F_3^*(t_f) = 30$ m).

The trajectory replanning is performed at time instant $t = 2.5$ s and it is combined with an FTC approach that consists in multiplying the applied thrust by $1/\delta_4$. The optimization problem (47) is

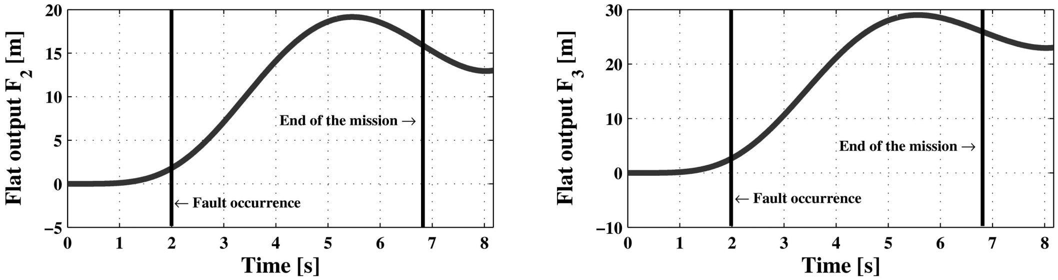

Fig. 8. Flat output F_2 and F_3 in fault case without accommodation.

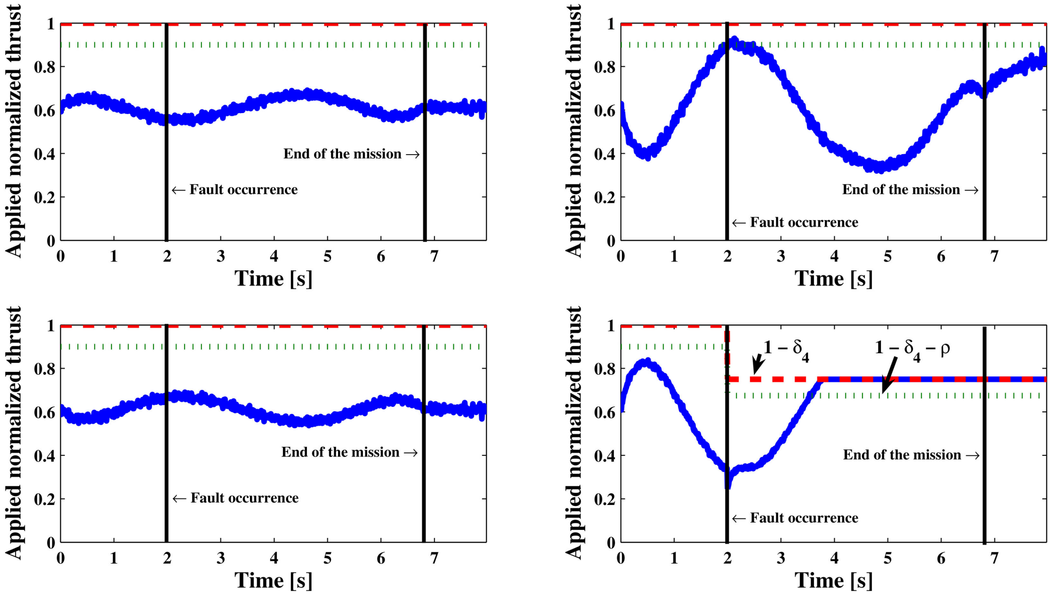

Fig. 9. Applied normalized thrusts $\underline{T}_i$ in fault case without accommodation.

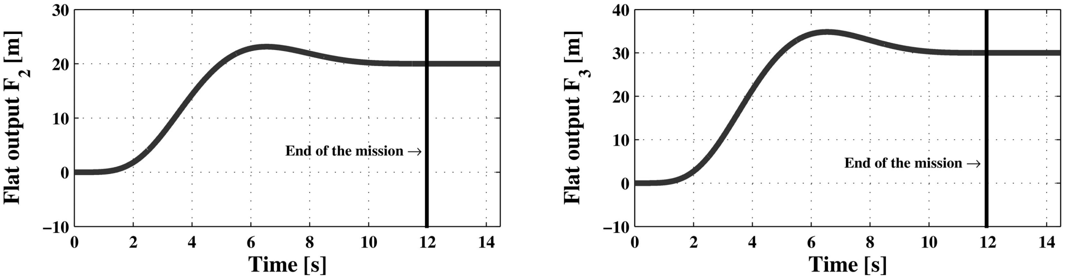

Fig. 10. Flat output F_2 and F_3 in fault case with accommodation.

solved by imposing $\rho = 0$ to get a feasible solution with $\delta_4 + \rho < 0.3875$. The solution obtained is t_f = 10.06 s.

It can be noted that the desired final position is attained with fault-tolerance and replanning (Fig. 10) and that the applied thrusts do not violate the rotor constraints (Fig. 11).

At the replanning instant $t = 2.5$ s, the velocity of the system is relatively big. The Bézier polynomial function used in the replanning is defined to verify the system conditions at the initial and the final time instants. However, the Bézier function does not allow to reshape the polynomial between the initial and the final time. This explains the overshoot in Fig. 10.

The application of the trajectory replanning to the nonlinear model (3) could not be done. This is because the linear model (6) employed to design the controller becomes invalid when faults occur. Faults destabilize the system and drive it out of the linear region ($z \neq$ constant and $\psi \neq 0$). One possible solution to this problem is to use the nonlinear model to design the system controller. This point will be investigated in our future work.

VI. DISCUSSION ON COMPUTING ISSUES

The objective of this study is to derive trajectory planning approaches that can be implemented for real-time applications. Thus, computing issues are

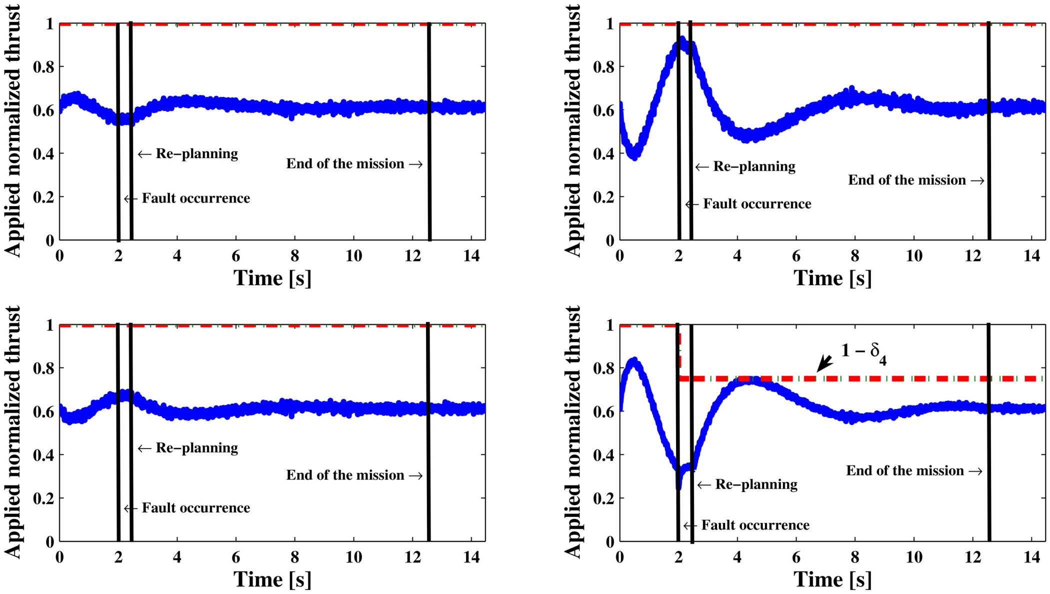

Fig. 11. Applied normalized thrusts $\underline{T}_i$ in fault case with accommodation.

TABLE I
Mean of the Computing Time over 10^2 Random Runs

Approach	Polynomial 3 for F_1/F_4 and 7 for F_2/F_3		Polynomial 5 for F_1/F_4 and 9 for F_2/F_3	
	Computing Time	Nominal Solution t_f	Computing Time	Nominal Solution t_f
Model (5) (No algebraic constraints)	23.67 s	7.44 s	60.44 s	6.88 s
Model (6) (No algebraic constraints)	3.21 s	7.52 s	5.71 s	6.67 s
Model (6) (With algebraic constraints)	0.0372 s	7.52 s	0.244 s	6.67 s

of great importance. By computing issues, we mean the computing load and the computing time. When calculations are to be performed at the ground station, one can allow heavy computing loads and large computing times. Once the calculations are done, the trajectory parameters can be communicated to the UAV.

However, if trajectory planning is to be performed on-board it is necessary to reduce both computing load and time. To obtain an idea about the computing time of our proposed approach, we have made 10^2 runs for the trajectory planning by using models (5) and (6). The trajectory planning is performed in the three-dimensional space with final conditions $F_1(t_f) = 4$ m, $F_2(t_f) = 20$ m, $F_3(t_f) = 30$ m, with $\rho = 0.1$ and all other terminal conditions are set to zero. To eliminate the role played by the initial guess in the computing time, the runs are performed with random initial guesses in an interval around the optimal solution.

The obtained results are summarized in Table I where three approaches are employed: model (5) without algebraic constraints and model (6) with and without algebraic constraints. Two cases are also compared. In the first case polynomials of degree 3 and 7 are used for F_1/F_4 and F_2/F_3, respectively. In the second case the polynomials are of degree 5 and 9, respectively (see (17) and (18)). It should be noted that algebraic constraints cannot be derived when model (5) is employed in the trajectory planning and thus this approach is not considered in the study.

By referring to the results obtained in Table I, one can note the following.

1) As previously argued in Section III, increasing the degree of the polynomials increases the computational load. Thus, with polynomial functions of degree 5 for F_1/F_4 and of degree 9 for F_2/F_3, it takes longer to get a solution. However, larger polynomial degrees allow for smoother control inputs and smaller t_f.

2) By comparing the computation times for models (5) and (6) without algebraic expressions, one can see that assuming hovering condition speeds up the solution by 7 to 10 times.

3) By comparing the last two rows in Table I, one can see the benefit in replacing the constraints of the optimization problem by their extrema where it takes less than 1 s to provide a solution. The extrema $T^*_{i_{\mathrm{Ext}}}$ are algebraic functions of t_0, t_f, and initial and final conditions. Thus, for trajectory planning it is sufficient to replace t_0 and the initial and final conditions with their numerical values to get the solution t_f of the optimization problem. This reduces the complexity of the optimization problem and the calculation requirements.

The above results are obtained by using MATLAB function fmincon and a Pentium M, 1.6 GHz PC with 768 MB of RAM. It is obvious that faster computing times can be obtained by using faster machines and software intended for real-time systems. Thus, real-time computation for the quadrotor UAV may be achieved with application-specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs).

VII. CONCLUSION

This paper investigates the problem of trajectory planning/replanning for a quadrotor UAV. Flatness techniques have been used to formulate the problem of trajectory planning into an optimization problem. The main advantage of flatness is that it allows to determine a priori if a specified trajectory is feasible and to change the trajectory parameters according to the system constraints. The objective of the trajectory planning approach is to find the profile of the reference trajectories (by tuning the time of the mission t_f) so that the system can be driven as fast as possible from one point to another without hitting actuator constraints. As can be seen in (28), the nominal control inputs u_i^* to be applied along the reference trajectories are polynomial functions. Thus, it can be noted that the resulting trajectories are the best trajectories that can be obtained by using flatness-based trajectory planning, but are not necessarily the optimal ones since it is possible to find faster trajectories by using some sort of bang-bang control [10]. This bang-bang control cannot be achieved by using flatness since the reference trajectories are a priori fixed to polynomials in (17) and (18) which in turn restrict u_i^* to polynomials. This issue may be seen as a limitation of this trajectory planning approach.

This work seeks to propose a trajectory planning/replanning technique that can be implemented on real-time applications. Thus, to reduce calculation requirements and computational load, the simplified model (6) is employed to replace the constraints of the optimization problem by their extrema. As shown in Section VI, replacing the constraints with their algebraic extrema drastically reduces the computation times. Using the simplified models in the trajectory planning increases the uncertainties and the mismatch with the system. Thus, future work will investigate how to reduce the complexity of the trajectory planning/replanning and the computational load when considering the nonlinear model (1) for the trajectory planning. To this end, flatness (with other trajectory parameterizations) and other techniques will be considered. Future work will also consider the application of the proposed approach to the quadrotor testbed at the Department of Mechanical and Industrial Engineering of Concordia University.

REFERENCES

[1] Henrion, D., Tarbouriech, S., and Kučera, V.
Control of linear systems subject to input constraints: A polynomial approach.
Automatica, **37** (2001), 597–604.

[2] Castelan, E., da Silva, Jr., J. G., and Cury, J.
A reduced-order framework applied to linear systems with constrained controls.
IEEE Transactions on Automatic Control, **41**, 2 (1996), 249–255.

[3] Tarbouriech, S., Garcia, G., and da Silva, Jr., J. G.
Robust stability of uncertain polytopic linear time-delay systems with saturating inputs: An LMI approach.
Computers and Electrical Engineering, **28** (2002), 157–169.

[4] Bemporad, A., Casavola, A., and Mosca, E.
Nonlinear control of constrained linear systems via predictive reference management.
IEEE Transactions on Automatic Control, **42**, 3 (1997), 340–349.

[5] Gilbert, E. and Kolmanovsky, I.
Nonlinear tracking control in the presence of state and control constraints: A generalized reference governor.
Automatica, **38**, 12 (2002), 2063–2073.

[6] Zhang, Y. M. and Jiang, J.
Fault tolerant control system design with explicit consideration of performance degradation.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 3 (2003), 838–848.

[7] Theilliol, D., Zhang, Y. M., and Ponsart, J-C.
Fault tolerant control system against actuator failures based on re-configuring reference input.
International Conference on Advances in Computational Tools for Engineering Applications (ACTEA'09), July 2009, pp. 480–485.

[8] Theilliol, D., Join, C., and Zhang, Y. M.
Actuator fault tolerant control based on a reconfigurable reference input.
International Journal of Applied Mathematics and Computer Science, **18**, 4 (2008), 553–560.

[9] Dardinier-Maron, V., Hamelin, F., and Noura, H.
A fault-tolerant control design against major actuators failures: Application to a three-tank system.
In *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, AZ, Dec. 1999, pp. 3569–3574.

[10] Lai, L-C., Yang, C-C., and Wu, C-J.
Time-optimal control of a hovering quad-rotor helicopter.
Journal of Intelligent and Robotic Systems, **45** (2006), 115–135.

[11] Meister, O., et al.
Adaptive path planning for a VTOL-UAV.
IEEE Aerospace and Electronic Systems Magazine, **24**, 7 (2009), 36–41.

[12] Kuwata, Y.
Trajectory planning for unmanned vehicles using robust receding horizon control.
Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[13] Mai, P., Join, C., and Reger, J.
Flatness-based fault-tolerant control of a nonlinear MIMO system using algebraic derivative estimation.
In *Preprints of the 3rd IFAC Symposium on System, Structure and Control*, Foz do Iguaçu, Brazil, Oct. 2007.

[14] Bouktir, Y., Haddad, M., and Chettibi, T.
Trajectory planning for a quadrotor helicopter.
16th Mediterranean Conference on Control and Automation, Ajaccio, Corsica, France, June 2008, pp. 1258–1263.

[15] Cowling, I., Whidborne, J., and Cooke, A.
Optimal trajectory planning and LQR control for a quadrotor UAV.
In *Proceedings of the UKACC International Conference on Control*, Glasgow, Scotland, Aug. 30–Sept. 11, 2006.

[16] Cowling, I., et al.
A prototype of an autonomous controller for a quadrotor UAV.
In *Proceedings of the European Control Conference*, Kos, Greece, July 2007, pp. 4001–4008.

[17] Noura, H., et al.
Fault-Tolerant Control Systems: Design and Practical Applications.
New York: Springer, 2009.

[18] Xu, R. and Ozguner, U.
Sliding mode control of a quadrotor helicopter.
In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, Dec. 2006, pp. 4957–4962.

[19] Waslander, S., et al.
Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning.
International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada, Aug. 2005, pp. 3712–3717.

[20] Omerdic, E. and Roberts, G.
Thruster fault diagnosis and accommodation for open-frame underwater vehicles.
Control Engineering Practice, **12** (2004), 1575–1598.

[21] Fliess, M., et al.
Flatness and defect of nonlinear systems: Introductory theory and example.
International Journal of Control, **61**, 6 (1995), 1327–1361.

[22] Hagenmeyer, V. and Delaleau, E.
Continuous-time non-linear flatness-based predictive control: An exact feedforward linearisation setting with an induction drive example.
International Journal of Control, **81**, 10 (2008), 1645–1663.

[23] Sira-Ramírez, H. and Agrawal, S.
Differentially Flat Systems.
New York: Marcel Dekker, 2004.

[24] Lévine, J. and Nguyen, D.
Flat output characterization for linear systems using polynomial matrices.
Systems and Control Letters, **48** (2003), 69–75.

[25] Mboup, M., Join, C., and Fliess, M.
Numerical differentiation with annihilators in noisy environment.
Numerical Algorithms, **50**, 4 (2009), 439–467.

[26] Levant, A. and Fridman, L.
Higher order sliding modes.
In W. Perruquetti and J. P. Barbot (Eds.), *Sliding Mode Control in Engineering*, New York: Marcel Dekker, 2002, pp. 53–101.

[27] Zoladek, H.
The topological proof of Abel-Ruffini theorem.
Journal of the Juliusz Schauder Center, **16** (2000), 253–265.

[28] Weisstein, E.
Abel's impossibility theorem.
MathWorld—A Wolfram Web Resource. [Online] available: http://mathworld.wolfram.com/AbelsImpossibilityTheorem.html.

Abbas Chamseddine received the degree of mechanical engineering from the Lebanese University in 2003, and the M.S. degree from the Ecole Centrale de Nantes, France, in 2004. He received his Ph.D. degree from Paul-Cézanne University, France, in 2007.

He was a Post-doc Fellow (PDF) at the Research Centre for Automatic Control of Nancy (CRAN) of Nancy-University, France before joining the Department of Mechanical and Industrial Engineering, Concordia University, as a PDF in September 2009. His research interests include fault diagnosis and fault-tolerant control for ground, aerial, and aerospace systems.

Dr. Chamseddine has published about 25 journal/conference papers and is co-author of a recently published book, *Fault-tolerant Control Systems: Design and Practical Applications*. He was an editorial board member of two national meetings for young researchers in 2008, France.

Youmin Zhang (M'99—SM'07) received his Ph.D. degree in 1995 from the Department of Automatic Control, Northwestern Polytechnical University, Xian, P.R. China.

He is currently an associate professor in the Department of Mechanical and Industrial Engineering, Concordia University, Canada. His main research interests and experience are in the areas of condition monitoring, fault diagnosis and fault-tolerant control systems; cooperative guidance, navigation and control of unmanned aerial/ground vehicles and spacecraft/satellites; dynamic systems modeling, estimation, identification and control; and advanced signal processing techniques for diagnosis, prognosis and health management of safety-critical systems and manufacturing processes.

Dr. Zhang has published 4 books, over 200 refereed journal and conference papers, including 58 journal papers. He is a senior member of AIAA, a member of IFAC Technical Committee on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS) and AIAA Infotech@Aerospace Program Committee on Unmanned Systems, and a member of Association for Unmanned Vehicle Systems International (AUVSI), Unmanned Systems Canada (USC), Canadian Aeronautics and Space Institute (CASI), and Canadian Society for Mechanical Engineering (CSME). He is an editorial board member of several international journals and an IPC member of many international conferences.

Camille Alain Rabbath received the Ph.D. degree in 1999 from McGill University.

He is currently defence scientist at Defence Research and Development Canada–Valcartier. He also holds an adjunct professorship position at Concordia University. He worked in industry from 1999 to 2002 in control systems design, and in modeling and simulation of aerospace and robotic systems. He is active in the development of cooperative unmanned aerial systems with health monitoring, and decision-making capabilities.

Dr. Rabbath is currently associate editor for the journals *IEEE Transactions on Control Systems Technology*, *Transactions of the Canadian Society for Mechanical Engineering*, and *Control Engineering Practice*.

Cédric Join received his Ph.D. in 2002.

He is an associate professor at University of Lorraine, France, and is a member of the INRIA project-team, ALIEN. His interest is in development of estimation techniques for linear and nonlinear systems. His research involves diagnosis, fault accommodation, and trajectory replanning.

In 2005, Dr. Join invented with M. Fliess, the notion of model-free control. His works are published in international journals (15), as book chapters (3), and in international conferences (50). His H-number equals 14 (following google scholar). With the collaboration of M. Fliess, M. Mboup, and H. Sira-Ramirez, 2 patents were published in 2006 concerning image and signal processing treatments and another one concerning hydroelectric power plants which won the innovation price of Ecole Polytechnique, Paris in 2010.

Didier Theilliol received the Ph.D. degree in control engineering from Nancy University, France, in 1993.

Since September 2004, he is a full professor in the Research Centre for Automatic Control of Nancy (CRAN), University of Lorraine, where he coordinates and leads national, European, and international R&D projects in steel industries, wastewater treatment plant, or aerospace domain. His current research interests include model-based fault diagnosis (FDI) method synthesis and active fault-tolerant control (FTC) system design for LTI, LPV, multi-linear systems and also reliability analysis.

Professor Theilliol has published over 100 journal/conference papers and is coauthor of a new book *Fault-tolerant Control Systems: Design and Practical Applications*.