Adaptive Trajectory Planning for a Quad-rotor Unmanned Aerial Vehicle

Abbas Chamseddine* and Youmin Zhang†

Concordia University, Montreal, Quebec H3G 1M8, Canada

Camille-Alain Rabbath‡

Defence Research and Development Canada, Valcartier, Quebec G3J 1X5, Canada

This work proposes an adaptive trajectory planning approach for a single quad-rotor unmanned aerial vehicle (UAV). The objective is to drive the system from an initial position to a final one without hitting actuator constraints while minimizing the total time of the mission. The approach employs differential flatness to represent the control inputs to be applied in function of the desired trajectories and formulates the trajectory planning problem as an optimization problem. An adaptive law is introduced to adapt the parameters of the optimization problem in function of the system uncertainties. This adaptation aims to avoid hitting the actuator constraints when the mismatch between the nominal model and the system is large.

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 can not 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 can not deliver the actuation inputs desired by the controller. Control systems are often linearly designed 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. Several works considered this problem either by avoiding saturation or allowing saturation while designing stabilizing controllers with saturating controls.

Other approaches have been investigated to deal with systems constraints: reference management or reference governor is proposed in the literature for systems with input and/or state related constraints. 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. A reference governor is designed for general discrete-time and continuous-time nonlinear systems with uncertainties. It relies on safety properties provided by sub-level sets of equilibria-parameterized functions.

A team of researchers at the Department of Mechanical and Industrial Engineering of Concordia University is currently working on the Networked Autonomous Vehicles (NAV) project to provide theoretical and experimental results on modeling, control, trajectory and path planning, formation flight, diagnosis and fault-tolerant control. In this context, this work considers the problem of adaptive trajectory planning for the quad-rotor UAV system. On the one hand, few research works treat the problem of trajectory planning...
for the quad-rotor UAV: a time-optimal motion-planning is proposed\(^7\) to generate a time-optimal motion between two configurations for a non-linear model of a hovering quad-rotor helicopter with four independently driven rotors. Several trajectory optimization algorithms are presented\(^8\) 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 quad-rotors. Flatness has been recently employed in trajectory planning for the quad-rotor UAV: a method is presented\(^9\) to generate time-optimal trajectories for the system. Cowling et al.\(^{10,11}\) present a quasi-optimal trajectory planner with a simple 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 in flatness-based trajectory planning\(^9-11\) is the parametrization of the trajectory: in some works,\(^9\) 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 other works\(^{10,11}\) several polynomial 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.

On the other hand, the term ‘adaptive path planning’ is employed in the context of obstacle avoidance;\(^{12,13}\) some 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.

In our work, the adaptive trajectory planning for the quad-rotor UAV system consists of two aspects. First, the differential flatness is employed to solve the problem of trajectory planning for the quad-rotor UAV with further investigation of the trajectory parametrization. This trajectory planning is performed only once at the start of the mission based on a deterministic model of the quad-rotor. However, due to model uncertainties, the applied control inputs may be larger than the nominal ones and the actuator constraints may be hit during the mission. Thus, in a second step, an adaptive law is employed to update the parameters of the trajectory in function of model uncertainties. To summarize, the contributions of this work are:

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 rotor constraints. The trajectory planning problem is formulated as an optimization problem.

2. An adaptive law is introduced to update the parameters of the optimization problem during flight. This adaptation aims to avoid hitting the actuator constraints in the presence of large model uncertainties.

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

### II. System Model

This work considers a commonly employed quad-rotor UAV model.\(^{14}\) Note that this is a simplified deterministic model of the more complex and uncertain real system:

\[
\begin{align*}
\ddot{x} &= u_1 (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) - \frac{k_1}{m} \dot{x} \\
\ddot{y} &= u_1 (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) - \frac{k_2}{m} \dot{y} \\
\ddot{z} &= -g + u_1 (\cos \phi \cos \theta) - \frac{k_3}{m} \dot{z}
\end{align*}
\]

\[
\begin{align*}
\ddot{\theta} &= u_2 - \frac{k_4}{J_1} \dot{\theta} \\
\ddot{\phi} &= u_3 - \frac{k_5}{J_2} \dot{\phi} \\
\ddot{\psi} &= u_4 - \frac{k_6}{J_3} \dot{\psi}
\end{align*}
\]

where \(x, y, z\) are the coordinates of the quad-rotor center of gravity in the earth-frame. \(\theta, \phi, \) and \(\psi\) are the roll, pitch and yaw angles respectively. \(m\) is the mass and \(J_i\) \((i = 1, 2, 3)\) are the moments of inertia along \(x, y, \) and \(z\) directions. \(k_i\) \((i = 1, \cdots, 6)\) are the drag coefficients and \(l\) is the distance from the center of gravity to each rotor. \(u_1\) is the total force applied to the quad-rotor in the \(z\)-direction. \(u_2, u_3, \) and \(u_4\) are
respectively the applied moments in \( \theta, \phi, \) and \( \psi \) directions. These force/moments are related to the thrusts generated by the rotors as follows:

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

(2)

where \( T_i \) \((i = 1, \ldots, 4)\) is the thrust generated by the \( i^{th} \) rotor and \( C \) is the thrust to moment scaling factor.

### A. Model Simplification

The controller and the trajectory planning approaches will be designed for the non-linear model (1) while assuming negligible drag coefficients at low speeds.\(^7\) Thus, (1) reads:

\[
\begin{align*}
\ddot{x} &= u_1 \left( \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \right) \\
\ddot{y} &= u_1 \left( \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \right) \\
\ddot{z} &= -g + u_1 \left( \cos \phi \cos \theta \right)
\end{align*}
\]

(3)

### B. Normalization

Before proceeding with the study, the vectors and the matrix of Equation 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.\(^{15}\) The first step is to find the maximum values (modules) of the inputs \( u_i \). It can be shown that with \( T_{\text{max}} = T_{\text{max}} = T_{\text{max}} = T_{\text{max}} \):

\[
\begin{align*}
u_{1,\text{max}} &= \frac{4}{m} T_{\text{max}}; & \nu_{2,\text{max}} &= \frac{2l}{J_1} T_{\text{max}}; & \nu_{3,\text{max}} &= \frac{2l}{J_2} T_{\text{max}}; & \nu_{4,\text{max}} &= \frac{2C}{J_3} T_{\text{max}}.
\end{align*}
\]

(4)

Dividing the vector components \( u_i \) by their maximum values, the normalized relation is then:

\[
\begin{pmatrix}
u_1 \\
\nu_{2,\text{max}} \\
\nu_{3,\text{max}} \\
\nu_{4,\text{max}}
\end{pmatrix} = \begin{pmatrix}
+\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} \\
-\frac{1}{2} & -\frac{1}{2} & +\frac{1}{2} & +\frac{1}{2} \\
-\frac{1}{2} & +\frac{1}{2} & +\frac{1}{2} & -\frac{1}{2} \\
+\frac{1}{2} & -\frac{1}{2} & +\frac{1}{2} & -\frac{1}{2}
\end{pmatrix}\begin{pmatrix}
T_1 \\
T_{\text{max}} \\
T_2 \\
T_{\text{max}} \\
T_3 \\
T_{\text{max}} \\
T_4 \\
T_{\text{max}}
\end{pmatrix}
\]

(5)

The normalized form (5) has a number of advantages compared to the standard form (2). First, the components \( T_i/T_{\text{max}} \) are dimensionless numbers, restricted to the standard interval \([-1, 1]\). This enables better understanding of the problem. Second, all physical parameters are removed from the matrix during the normalization process. This simplifies the calculations. The inverse of (5) is a simple expression and it is:

\[
\begin{pmatrix}
T_1 \\
T_2 \\
T_3 \\
T_4
\end{pmatrix} = \begin{pmatrix}
1 & -\frac{1}{2} & -\frac{1}{2} & +\frac{1}{2} \\
1 & +\frac{1}{2} & +\frac{1}{2} & -\frac{1}{2} \\
1 & +\frac{1}{2} & +\frac{1}{2} & +\frac{1}{2} \\
1 & +\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2}
\end{pmatrix}\begin{pmatrix}
\nu_1 \\
\nu_{2,\text{max}} \\
\nu_{3,\text{max}} \\
\nu_{4,\text{max}}
\end{pmatrix}
\]

(6)
III. Flatness-based Control

A flatness-based controller will first be designed for the quad-rotor system to track the desired reference trajectories. The flatness property is described as follows.\textsuperscript{16,17} A dynamical system:

\[
\begin{align*}
\dot{x} &= f(x, u) \\
\dot{y} &= h(x),
\end{align*}
\]

with \( x \in \mathbb{R}^n \) and \( u \in \mathbb{R}^m \), is flat if and only if there exist variables \( F \in \mathbb{R}^m \) called the flat outputs such that:

\[
x = \Xi_1(F, \dot{F}, \cdots, F^{(n-1)}), \quad y = \Xi_2(F, \dot{F}, \cdots, F^{(n-1)}), \quad u = \Xi_3(F, \dot{F}, \cdots, F^{(n)}).
\]

\( \Xi_1, \Xi_2, \) and \( \Xi_3 \) are three smooth mappings and \( F^{(i)} \) is the \( i \)-th derivative of \( F \).

The determination of the flat outputs is investigated for SISO and MIMO linear and nonlinear systems\textsuperscript{18} and for linear system in polynomial matrix form.\textsuperscript{19} 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 \( \psi \), the parametrization of \( \theta \) and \( \phi \) in function of the flat outputs is:

\[
\begin{align*}
\theta &= \tan \left( \frac{\cos F_4 \dot{F_2} + \sin F_4 F_3}{\dot{F_1} + g} \right); & \phi &= \tan \left( \frac{\sin F_4 \dot{F_2} - \cos F_4 \dot{F_3}}{\dot{F_1} + g} \right).
\end{align*}
\]

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

\[
u_1 = \frac{\dot{F}_4 + g}{\cos \phi \cos \theta}; \quad u_2 = \ddot{\theta}; \quad u_3 = \dddot{\phi}; \quad u_4 = \dddot{\bar{F}}_4,
\]

where \( \theta \) and \( \phi \) are given in (8) and \( \ddot{\theta} \) and \( \dddot{\phi} \) are function of the flat outputs and can be derived from \( \theta \) and \( \phi \).

From the differential parametrization (9), a flatness-based tracking control law can be obtained when replacing \( \ddot{F}_1, \dddot{F}_2, \dddot{F}_3, \) and \( \dddot{F}_4 \) by the auxiliary control inputs \( \ddot{u}_1, \ddot{u}_2, \dddot{u}_3, \) and \( \dddot{u}_4 \) respectively in \( u_1, u_2, u_3, \) and \( u_4 \), hence:

\[
\ddot{F}_1 = \ddot{u}_1; \quad F_2^{(4)} = \ddot{u}_2; \quad F_3^{(4)} = \dddot{u}_3; \quad \dddot{F}_4 = \dddot{u}_4,
\]

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

\[
\begin{align*}
\ddot{u}_1 &= \dddot{F}_1^* + K_{11}(\dddot{F}_1^* - \dddot{F}_1) + K_{12} (F_1^* - F_1) \\
\ddot{u}_2 &= F_2^{(4)*} + K_{21}(F_2^{(4)*} - F_2^{(3)*}) + K_{22}(\dddot{F}_2 - \dddot{F}_2) + K_{23}(\dddot{F}_3 - \dddot{F}_3) + K_{24} (F_2^* - F_2) \\
\dddot{u}_3 &= F_3^{(4)*} + K_{31}(F_3^{(4)*} - F_3^{(3)*}) + K_{32}(\dddot{F}_3 - \dddot{F}_3) + K_{33}(\dddot{F}_3^* - \dddot{F}_3) + K_{34} (F_3^* - F_3) \\
\dddot{u}_4 &= \dddot{F}_4^* + K_{41}(\dddot{F}_4^* - \dddot{F}_4) + K_{42} (F_4^* - F_4)
\end{align*}
\]

ensures the tracking errors \( e_i = F_i^* - F_i \ (i = 1, \cdots, 4) \) 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:

\[
u_1 = \frac{\ddot{u}_1 + g}{\cos \phi \cos \theta}; \quad u_2 = \ddot{\theta}; \quad u_3 = \dddot{\phi}; \quad u_4 = \dddot{u}_4,
\]

where \( \ddot{\theta} \) is obtained when substituting \( F_2^{(4)} \) with \( \dddot{u}_2 \) and \( \dddot{\phi} \) is obtained when substituting \( F_3^{(4)} \) with \( \dddot{u}_3 \). \( u_i \ (i = 1, \cdots, 4) \) are function of the derivatives of the flat outputs. These can be derived by using for example an algebraic derivative estimation\textsuperscript{20} or sliding mode differentiation\textsuperscript{21} when they are not accessible via the measurements. It should also be noted that the measurements needed to accomplish the controller given in (12) are the cartesian coordinates \( x, y, z, \) and the yaw angle \( \psi \) and their derivatives to some fixed order. The measurements of pitch \( \theta \) and roll \( \phi \) angles are not needed since they can be expressed in function of the flat outputs (Equation 8).
Several methods can be used to design the reference trajectories $F_i^*$. In this work we employ the Bézier polynomial function. This is because the coefficients of the polynomial can be easily calculated in function of the initial and the terminal conditions. A general Bézier polynomial function of degree $n$ is:

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

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

For our quad-rotor UAV, it can be seen in Equation 9 that the relative degrees are $r_1 = 2$, $r_2 = 4$, $r_3 = 4$, and $r_4 = 2$. Thus, the system imposes to employ a Bézier polynomial function of degree 3 for $F_1$ and $F_4$ and a Bézier polynomial function of degree 7 for $F_2$ and $F_3$. However, to obtain smoother control inputs we will 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:

$$F_i^* = a_3 t^5 + a_4 t^4 + a_5 t^3 + a_6 t^2 + a_7 t + a_8 \quad (i = 1, 4),$$

and

$$F_i^* = a_0 t^9 + a_1 t^8 + a_2 t^7 + a_3 t^6 + a_4 t^5 + a_5 t^4 + a_6 t^3 + a_7 t^2 + a_8 t + a_9 \quad (i = 2, 3).$$

The coefficients $a_i$ ($i = 1, 4$ and $j = 0, \cdots, 5$) are calculated to verify the initial conditions $F_i(t_0)$, $F_i'(t_0)$, $F_i''(t_0)$ and the final conditions $F_i(t_f)$, $F_i'(t_f)$, $F_i''(t_f)$. The coefficients $a_j$ ($i = 2, 3$ and $j = 0, \cdots, 9$) are calculated to verify the initial conditions $F_i(t_0)$, $F_i'(t_0)$, $F_i''(t_0)$, $F_i'''(t_0)$, $F_i^{(4)}(t_0)$ and the final conditions $F_i(t_f)$, $F_i'(t_f)$, $F_i''(t_f)$, $F_i'''(t_f)$, $F_i^{(4)}(t_f)$. $t_0$ and $t_f$ are respectively the initial and the final instants of the mission. $t_0$ is usually set to 0.

Note that smooth trajectories can be obtained with a polynomial function of degree 5 for $F_1$ and $F_4$ and a polynomial function of degree 9 for $F_2$ and $F_3$. This is because it is possible to impose zero velocities and accelerations at $t_0$ and $t_f$ for $F_1$ and $F_4$. It is also possible to impose zero $\dot{F}$, $\ddot{F}$, $F^{(3)}$, $F^{(4)}$ at $t_0$ and $t_f$ for $F_2$ and $F_3$. In the general case, the degrees of the polynomial function are selected depending on the initial and final conditions to be verified along the trajectory.

### IV. Flatness-based Trajectory Planning

The basic idea of the flatness-based trajectory planning is to represent the control inputs to be applied in function of the trajectory to be followed. This allows then to tune the parameters of the desired trajectory so that the actuator constraints are not violated. The nominal control inputs $u_i^*$ to be applied for the nominal reference trajectories $F_i^*$ are given by:

$$u_1^* = \frac{\ddot{F}_1^* + g}{\cos\phi^* \cos\theta^*}; \quad u_2^* = \dot{\phi}^*; \quad u_3^* = \dot{\theta}^*; \quad u_4^* = \ddot{p}_z^*,$$

where

$$\phi^* = \frac{\cos F_4^* \ddot{F}_4^* + \sin F_4^* \dot{F}_4^*}{F_4^* + g}; \quad \theta^* = atan \left( \frac{\cos F_4^* \ddot{F}_4^* + \sin F_4^* \dot{F}_4^*}{F_4^* + g} \right).$$

Let us denote $T_i = T_i/T_{\text{max}}$. According to Equation 6, one can note that the nominal thrusts $T_i^*$ to be applied along the reference trajectories $F_i^*$ are:

$$\begin{align*}
T_1^* &= \frac{u_1^*}{u_{1\text{max}}} - \frac{1}{2} \frac{u_2^*}{u_{2\text{max}}} - \frac{1}{2} \frac{u_3^*}{u_{3\text{max}}} + \frac{1}{2} \frac{u_4^*}{u_{4\text{max}}} \\
T_2^* &= \frac{u_1^*}{u_{1\text{max}}} - \frac{1}{2} \frac{u_2^*}{u_{2\text{max}}} + \frac{1}{2} \frac{u_3^*}{u_{3\text{max}}} - \frac{1}{2} \frac{u_4^*}{u_{4\text{max}}} \\
T_3^* &= \frac{u_1^*}{u_{1\text{max}}} + \frac{1}{2} \frac{u_2^*}{u_{2\text{max}}} + \frac{1}{2} \frac{u_3^*}{u_{3\text{max}}} + \frac{1}{2} \frac{u_4^*}{u_{4\text{max}}} \\
T_4^* &= \frac{u_1^*}{u_{1\text{max}}} + \frac{1}{2} \frac{u_2^*}{u_{2\text{max}}} - \frac{1}{2} \frac{u_3^*}{u_{3\text{max}}} - \frac{1}{2} \frac{u_4^*}{u_{4\text{max}}}.
\end{align*}$$

American Institute of Aeronautics and Astronautics
where \( u_{i_{\text{max}}} \) are constants and are given in (4) and \( u^*_i \) are the nominal control inputs and are given in (16). The trajectory planning problem consists in determining the minimal time to drive the system from an initial state to a final state without violating the constraints of the UAV rotors. Mathematically, this can be expressed as an optimization problem:

\[
P \left\{ \begin{array}{l}
\text{Minimize} & t_f \\
\text{Subject to} & T^*(t) \in \Omega
\end{array} \right.
\]

(19)

where \( T^* = [T_1^* \ T_2^* \ T_3^* \ T_4^*]^T \) and \( \Omega \) is the domain of attainable thrusts. The domain \( \Omega \) is constituted of all the values that \( T^*_i \) can take \((i = 1, \cdots, 4)\). It is clear that \( \Omega \) is a tesseract or a 4-cube. It is defined as:

\[
\Omega = \{ T \in \mathbb{R}^4 : 0 \leq T_i \leq 1 \} \subset \mathbb{R}^4.
\]

(20)

Two methods can be employed to represent the constraint of the optimization problem (19). The first representation consists in imposing that each nominal thrust to be applied is less than 1 \((T_i \in [0, 1])\) after normalization in Section B). Thus, problem (19) reads:

\[
P_1 \left\{ \begin{array}{l}
\text{Minimize} & t_f \\
\text{Subject to} & 0 \leq T^*_i(t) \leq 1 ; \ i = 1, \cdots, 4
\end{array} \right.
\]

(21)

The second representation consists in imposing that the nominal thrusts remain inside the largest sphere included in \( \Omega \). In two dimensional space \( \mathbb{R}^2 \), the largest disk included in a unit square is the disk of origin \((0.5,0.5)\) and of radius 0.5. In three dimensional space \( \mathbb{R}^3 \), the largest insphere included in the unit cube is the sphere of origin \((0.5,0.5,0.5)\) and of radius 0.5. By analogy, the largest insphere included in the 4-unit cube \( \Omega \) is the 4D-sphere of origin \((0.5,0.5,0.5,0.5)\) and of radius 0.5. Thus, the second representation reads:

\[
P_2 \left\{ \begin{array}{l}
\text{Minimize} & t_f \\
\text{Subject to} & \sum_{i=1}^{4} (T^*_i(t) - 0.5)^2 \leq 0.5^2
\end{array} \right.
\]

(22)

where the left-hand side member of the constraint of \( P_2 \) is the squared distance from the origin of the 4D-sphere to the nominal thrusts. This distance is to be smaller than the radius \((R = 0.5)\) of the 4D-sphere. According to the definition of \( T^*_i \) in (18), it can be shown that:

\[
\sum_{i=1}^{4} (T^*_i(t) - 0.5)^2 = 1 - 4u^*_1(t) + 4u^*_2(t) + u^*_3(t) + 3u^*_4(t).
\]

(23)

where \( u^*_i = u^*_i/u_{i_{\text{max}}} \). Thus, the optimization problem (22) can be written as:

\[
P_2 \left\{ \begin{array}{l}
\text{Minimize} & t_f \\
\text{Subject to} & 1 - 4u^*_1(t) + 4u^*_2(t) + u^*_3(t) + 3u^*_4(t) \leq 0.5^2
\end{array} \right.
\]

(24)

It is clear that the 4-D sphere is smaller than the 4-cube \( \Omega \) and thus the allowable thrusts of problem (24) are more constrained than those of problem (21). However, it is worth to investigate both representations (21) and (24) from calculation requirements and convexity point of view.

The above study is carried out using the nominal control input (16). However, in practice, an additional control term is used to tackle uncertainties and to ensure the stability of the closed-loop system (see Equation 11). To take into consideration the control input generated by the additional control term, a safety margin is created by introducing a constant \( \rho \) in the optimization problems (21) and (24) as follows:

\[
P_1 \left\{ \begin{array}{l}
\text{Minimize} & t_f \\
\text{Subject to} & 0 \leq T^*_i(t) \leq 1 - \rho ; \ i = 1, \cdots, 4
\end{array} \right.
\]

(25)

and

\[
P_2 \left\{ \begin{array}{l}
\text{Minimize} & t_f \\
\text{Subject to} & 1 - 4u^*_1(t) + 4u^*_2(t) + u^*_3(t) + 3u^*_4(t) \leq (1 - \rho)^2 \ 0.5^2
\end{array} \right.
\]

(26)
where \( 0 \leq \rho < 1 \) depending on the model uncertainties. It is clear that the choice of \( \rho \) requires to quantify model uncertainties which is not an easy task. This point will be investigated in the next section via the adaptive trajectory planning.

As can be seen, the constraints of the optimization problems (25) and (26) are function of the time \( t \). It can be noted that in some cases, it is possible to replace a constraint by its optimum to eliminate \( t \). This can be performed when the constraints are simple functions of \( t \). However, it can be noted that the constraints for the trajectory planning of the six degrees of freedom nonlinear system are complex functions of \( t \) and thus optimum cannot be derived. In this work, we propose a simple method to solve the optimization problems (25) and (26). This method consists in splitting the time interval \([t_0, t_f]\) into \( N \) intervals and in imposing that the constraints are verified at each of the \( N + 1 \) boundary points. In this case, the trajectory planning optimization problems can be formulated as follows:

\[
P_1 \left\{ \begin{array}{l}
\text{Minimize} & \ t_f \\
\text{Subject to} & \ 0 \leq T^*_i(kt_f/N) \leq 1 - \rho \ ; \ i = 1, \ldots, 4 \ ; \ k = 0, \ldots, N ,
\end{array} \right.
\] (27)

and

\[
P_2 \left\{ \begin{array}{l}
\text{Minimize} & \ t_f \\
\text{Subject to} & \ 1 - 4u_{i1}^2(kt_f/N) + 4u_{i1}^2(kt_f/N) + u_{i2}^2(kt_f/N) + u_{i2}^2(kt_f/N) \\
& + u_{i2}^2(kt_f/N) \leq (1 - \rho)^2/0.5^2 \ ; \ k = 0, \ldots, N ,
\end{array} \right.
\] (28)

It is clear that the larger \( N \) is, the more affine the solution is but the heavier the calculations are. When the number of intervals \( N \) is small, the constraints will be verified at the \( N + 1 \) boundary points. However, it is not guaranteed that the constraints will be verified along the intervals. One solution to this problem is to impose an upper bound for the rate of change of the constraints. The objective is then to limit the rate of change of the constraints, as for example:

\[
\frac{dT^*_i(t)}{dt} \leq \frac{2\rho}{\Delta t}
\] (29)

and

\[
\frac{dC^*_i(t)}{dt} \leq \frac{0.5^2(2 - \rho)\rho}{\Delta t}
\] (30)

where \( \Delta t = t_f/N \) and \( C^*_i(t) \) is the constraint of the second representation (26), i.e. \( C^*_i(t) = 1 - 4u_{i1}^2(t) + 4u_{i1}^2(t) + u_{i2}^2(t) + u_{i2}^2(t) \). Figure 1 illustrates the idea of imposing an upper bound on the rate of change of the constraints. For a small \( N \) (and a fixed \( t_f \)), \( \Delta t \) is large and the upper bound for the allowable rate of change of the constraints given in (29) and (30) is small and vice-versa. Thus, imposing an upper bound for the rate of change of the constraints helps in avoiding to hit actuator constraints on the intervals \([k\Delta t, (k + 1)\Delta t] \) for \( k = 0, \ldots, N - 1 \) specially when \( N \) is small.

\[ \text{Figure 1. Schematic representation of the constraints rate of change} \]

By introducing the upper bounds on the rate of change of the constraints in the optimization problems (27) and (28) at each of the \( N + 1 \) boundary points, one obtains:

\[
P_1 \left\{ \begin{array}{l}
\text{Minimize} & \ t_f \\
\text{Subject to} & \ 0 \leq T^*_i(kt_f/N) \leq 1 - \rho \ ; \ i = 1, \ldots, 4 \ ; \ k = 0, \ldots, N , \\
& \ T^*_i(kt_f/N) \leq 2\rho N/t_f
\end{array} \right.
\] (31)

American Institute of Aeronautics and Astronautics
On the one hand, if $\rho$ uncertainties. Model uncertainties are caused by the mismatch between the mathematical model and the physical system due to unmodeled or neglected dynamics, simplification of highly nonlinear behavior and inaccurate values of the physical parameters (mass, inertia, drag coefficients etc.).

On the other hand, if $\rho$ (26). The parameter $\rho$ is chosen to be small, this increases the domain of allowable thrusts. In this case, the safety margin is small and the tendency to hit the actuator constraints is larger. However, this reduces the domain of allowable thrusts and increases the mission time $t_f$. Thus, it is important to find an adequate $\rho$ so that the actuator constraints are not violated and the mission time is not very large.

The idea in this section is to use an adaptive law to tune the parameter $\rho$ during the mission rather than fixing $\rho$ at a constant value. The principle of the adaptive trajectory planning is depicted in Figure 2. The adaptive law updates the parameter $\rho$ of the optimization problem to modify the reference trajectory $F^*$ in function of the generated control inputs. This adaptation is to be done on-line during the mission.

![Figure 2. Principle of the adaptive trajectory planning](image)

During a specified mission, the applied control input $u_i$ to the system which is generated by the controller is the sum of the nominal control input $u_i^*$ and an additional term $\Delta u_i$ due to model uncertainties:

$$u_i = u_i^* + \Delta u_i.$$  \hfill (33)

According to Equation 6, the applied thrusts are:

$$T = M u = M u^* + M \Delta u,$$  \hfill (34)

where $M$ is the matrix in Equation 6, $T = [T_1, T_2, T_3, T_4]^T$, $u = [u_1, u_2, u_3, u_4]^T$, $u^* = [u_1^*, u_2^*, u_3^*, u_4^*]^T$, and $\Delta u = [\Delta u_1, \Delta u_2, \Delta u_3, \Delta u_4]^T$. 

The computing issues and the comparison between the two representations of the optimization problems will be investigated in the simulation section. The next section addresses the adaptive trajectory planning problem to deal with model uncertainties.

V. Adaptive Trajectory Planning

The previous section considers the problem of trajectory planning with the objective to move the system from an initial configuration to a final one as fast as possible while avoiding to hit the actuator constraints. This problem is formulated as an optimization problem where two representations are proposed in (25) and (26). The parameter $\rho$ is introduced to create a safety margin to handle the control inputs caused by model uncertainties. Model uncertainties are caused by the mismatch between the mathematical model and the physical system due to unmodeled or neglected dynamics, simplification of highly nonlinear behavior and inaccurate values of the physical parameters (mass, inertia, drag coefficients etc.).

Solving the optimization problem for the trajectory planning is performed only once at the beginning of the mission. The parameter $\rho$ should be chosen according to model uncertainties which is not an easy task. On the one hand, if $\rho$ is chosen to be small, this increases the domain of allowable thrusts. In this case, the time of the mission $t_f$ is relatively small. However, the safety margin is small and the tendency to hit the actuator constraints is larger. On the other hand, if $\rho$ is chosen to be large, the safety margin is large and the tendency to hit the actuator constraints is smaller. However, this reduces the domain of allowable thrusts and increases the mission time $t_f$. Thus, it is important to find an adequate $\rho$ so that the actuator constraints are not violated and the mission time is not very large.

The idea in this section is to use an adaptive law to tune the parameter $\rho$ during the mission rather than fixing $\rho$ at a constant value. The principle of the adaptive trajectory planning is depicted in Figure 2. The adaptive law updates the parameter $\rho$ of the optimization problem to modify the reference trajectory $F^*$ in function of the generated control inputs. This adaptation is to be done on-line during the mission.
For the first representation of the trajectory planning problem \( P_1 \), the applied thrust \( T_i \) (\( i = 1, \cdots, 4 \)) is allowable (i.e. can be generated by the actuator) if:

\[ T_i = m_i \Delta u + m_i \Delta u \leq 1, \tag{35} \]

where \( m_i \) is the \( i^{th} \) row of matrix \( M \) (\( i = 1, \cdots, 4 \)). For simplicity, we will assume\(^a\) that \( N \) in problem (27) is sufficiently large so that the nominal thrusts:

\[ T^\ast_i = m_i \Delta u^\ast \leq 1 - \rho. \tag{36} \]

Equation 36 is ensured by the optimization problem and thus, Equation 35 holds if:

\[ m_i \Delta u \leq \rho. \tag{37} \]

Similarly as for the first representation of the trajectory planning problem \( P_1 \), in the second representation \( P_2 \), the applied thrusts \( T_i \) are allowable if they are contained in the unit 4D-sphere of origin (0.5,0.5,0.5,0.5) and of radius 0.5:

\[ \sum_{i=1}^{4} (T_i - 0.5)^2 \leq 0.5^2. \tag{38} \]

Substituting (6) and (33) in (38), one obtains that the applied thrusts are contained in the unit 4D-sphere if:

\[ 1 - 4u_1^* - 4\Delta u_1 + 4u_1^2 + 8u_1^* \Delta u_1 + 4(\Delta u_1)^2 + u_2^* + 2u_2^* \Delta u_2 + (\Delta u_2)^2 + u_2^2 + 2u_2^* \Delta u_3 + (\Delta u_3)^2 + u_3^2 + 2u_3^* \Delta u_3 + (\Delta u_3)^2 \leq 0.5^2. \tag{39} \]

Similarly as before, we will assume that \( N \) in the optimization problem (28) is sufficiently large so that the nominal thrusts:

\[ \sum_{i=1}^{4} (T_i^\ast(t) - 0.5)^2 = 1 - 4u_1(t) + 4u_1^2(t) + u_2^2(t) + u_3^2(t) + u_4^2(t) \leq (1 - \rho)^20.5^2. \tag{40} \]

Thus, Equation 39 holds if:

\[ -4\Delta u_1 + 8u_1^* \Delta u_1 + 4(\Delta u_1)^2 + 2u_2^* \Delta u_2 + (\Delta u_2)^2 + 2u_2^* \Delta u_3 + (\Delta u_3)^2 + 2u_3^* \Delta u_3 + (\Delta u_3)^2 \leq \rho(2 - \rho)0.5^2. \tag{41} \]

As a conclusion, the applied thrusts can be generated by the actuators if (37) and (41) are verified during the mission. One way to ensure that is to update \( \rho \) in function of model uncertainties. Clearly, model uncertainties are not easy to measure, access or evaluate. However, we have access to the left-hand sides of Equations 37 and 41 since they are generated by the controller. These terms reflect the amplitude of model uncertainties and will be employed to tune \( \rho \) so that (37) and (41) are verified.

It is obvious that for \( 0 \leq \rho \leq 1 \), one should pick the minimal \( \rho \) that verifies (37) (resp. (41)) when the first (resp. second) representation is employed. Let \( \rho_1 \) and \( \rho_2 \) be the safety margins issued from the first and the second representation respectively. Consider first Equation 37, in this case \( \rho_1 \) can be chosen as follows:

\[ \rho_1(t) = \begin{cases} 0 & \text{if } m_i \Delta u \leq 0 \\ m_i \Delta u & \text{if } m_i \Delta u > 0 \end{cases} \tag{42} \]

and

\[ \rho_1(t) = \max(\rho_1(t)) ; \ i = (1, \cdots, 4). \tag{43} \]

Consider now Equation 41 and define \( \Lambda \) as the left-hand side member, i.e. \( \Lambda = -4\Delta u_1 + 8u_1^* \Delta u_1 + 4(\Delta u_1)^2 + 2u_2^* \Delta u_2 + (\Delta u_2)^2 + 2u_2^* \Delta u_3 + (\Delta u_3)^2 + 2u_3^* \Delta u_3 + (\Delta u_3)^2 \). If \( \Lambda > 0 \), then \( \rho_2 \) is the solution of:

\[ \rho_2^2 - 2\rho_2 + 4\Lambda \leq 0. \tag{44} \]

\(^a\)The study can be easily carried out for the case when \( N \) is small and that problem (31) is used instead for the trajectory planning.
At the limit, $\rho_2$ is the solution of:

$$\rho_2^2 - 2\rho_2 + 4\Lambda = 0. \quad (45)$$

Equation 45 admits a solution if the discriminant $\Delta = 4 - 16\Lambda \geq 0$. In other words, if $\Lambda \leq 0.5^2$ that corresponds to the squared radius of the sphere. In this case, two solutions can be obtained:

$$\rho_2(t) = \begin{cases} 1 + \sqrt{1 - 4\Lambda} & \text{if } \Lambda \leq 0 \\ 1 - \sqrt{1 - 4\Lambda} & \text{if } \Lambda > 0 \end{cases} \quad (46)$$

The first solution is larger than 1 and thus can not be considered as $0 \leq \rho_2 \leq 1$. The square root of the second solution can be approximated by using Maclaurin series expansion up to order 4 and thus, $\rho_2$ can be chosen as follows:

$$\rho_2(t) = \begin{cases} 0 & \text{if } \Lambda \leq 0 \\ 2\Lambda + 2\Lambda^2 + 4\Lambda^3 + 10\Lambda^4 & \text{if } \Lambda > 0 \end{cases} \quad (47)$$

According to (43) and (47), a value for $\rho_i \ (i = 1, 2)$ is obtained at each step time. A simple method to update $\rho$ is to take the maximal value of $\rho_i$ over the interval $[t_0, t_f]$:

$$\rho = \max(\rho_i(t)) \ ; \ i = 1, 2. \quad (48)$$

While $\rho$ is updated at each step time, the adaptive planning however can be performed at fixed instants or when $\rho$ changes in certain amount. This trajectory adaptation is to be performed on-line during the mission. Thus, it is necessary to solve the trajectory planning optimization problem in real-time. For this purpose, the computing time issues are investigated in the next section.

**VI. 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. The numerical values of the parameters are given in Table 1.

<table>
<thead>
<tr>
<th>Nomenclature</th>
<th>Numerical value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>2 kg</td>
</tr>
<tr>
<td>$J_1, J_2, J_3$</td>
<td>1.25, 1.25, 2.50 $Ns^2/rad$</td>
</tr>
<tr>
<td>$l$</td>
<td>0.2 m</td>
</tr>
<tr>
<td>$k_i \ (i = 1, 2, 3)$</td>
<td>0.01 $Ns/m$</td>
</tr>
<tr>
<td>$k_i \ (i = 4, 5, 6)$</td>
<td>0.012 $Ns/rad$</td>
</tr>
<tr>
<td>$g$</td>
<td>9.8 $m/s^2$</td>
</tr>
<tr>
<td>$T_{max}$</td>
<td>8 N</td>
</tr>
</tbody>
</table>

**A. Flatness-based Control**

This simulation tests for the flatness-based controller developed in Section III. The quad-rotor is required to take-off from an initial configuration and then to go to a final configuration. All the initial conditions are zero. The final conditions are $F_1 = 4 \ m$, $F_2 = 200 \ m$, $F_3 = 300 \ m$, $F_4 = 0.5 \ rad$ and zero velocities and accelerations. These initial and final conditions will be employed in the subsequent sections. For illustration, the time of the mission is set to $t_f = 20 \ s$.

Figure 3 shows the time evolution of the four flat outputs of the system. It is clear that the proposed controller is able to drive the system from the desired configuration to the final one.

Figure 4 shows the applied thrusts to the system. They are smooth at the beginning and the end of the mission. They are also non zero after the end of the mission to keep the system in hovering position. It is obvious that the system can accomplish the mission in less than 20 s since the applied thrusts are not very close to $T_{max} = 1$. This point will be addressed in the next section.
B. Four-directional Trajectory Planning

Before testing the four-directional trajectory planning, we give in Table 2 a comparison between the two representations of the optimization problem given in Section IV. The comparison is carried out for two cases: without and with bounds on the rate of change of the constraints. In each case, we give the optimal solution (in seconds), the mean of the computing times (in seconds) and the percentage of converged solutions for some values of $N$. To calculate the mean, only the computing times of the converged solutions are considered. The study is carried out with 100 runs and with $\rho = 0.1$. To eliminate the role played by the initial guess in the computing times, the 100 runs are performed with fixed step initial guesses over the interval [1, 30].
Table 2. Comparison between the two representations

<table>
<thead>
<tr>
<th>N</th>
<th>Solution Without bounds</th>
<th>Solution With bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Percentage</td>
</tr>
<tr>
<td>1st representation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>18.47</td>
<td>0.177</td>
</tr>
<tr>
<td>50</td>
<td>18.49</td>
<td>0.759</td>
</tr>
<tr>
<td>100</td>
<td>18.51</td>
<td>1.941</td>
</tr>
<tr>
<td>2nd representation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>23.41</td>
<td>0.215</td>
</tr>
<tr>
<td>50</td>
<td>23.42</td>
<td>0.827</td>
</tr>
<tr>
<td>100</td>
<td>23.45</td>
<td>1.616</td>
</tr>
</tbody>
</table>

The results given in Table 2 can be explained as follows:

- In all the cases, the solutions obtained with the second representation are greater than those obtained with the first representation. This is because the domain in the former case is smaller than the one in the latter case.

- When \( N \) is small, the solution with bounds on the rate of change of the constraints is larger than that obtained without bounds. This is because for small \( N \), the bound is small which increases the time of the mission \( t_f \) (see Equation 31).

- When \( N \) is increased, the solution is the same with and without bounds. Thus, as argued in Section IV, it is not necessary to impose bounds on the constraints when \( N \) is relatively large.

- No improvements in the computing times or the percentages of converged solutions are obtained when bounds are imposed on the rate of change of the constraints. Thus, with a sufficiently large \( N \), the bounds seem to be useless for our application.

- Without bounds on the constraints, the comparison between the two representation does not reveal an obvious advantage of one over the other in terms of computing times and percentage of converged solutions.

It should be noted that the above results are obtained by using MATLAB function \textit{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 associated with high complex planning for the non-linear 6 degrees of freedom may be achieved with application-specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs).

Let us apply now flatness-based trajectory planning to the quad-rotor UAV. According to Table 2, for \( N = 100, \rho = 0.1 \) and with the first representation of the optimization problem, the minimal time \( t_f \) to drive the system from its initial position to the final desired one while not hitting actuator constraints is \( t_f = 18.51 \) s. Figure 5 shows the applied thrusts when trajectory planning is employed. The evolution of the flat outputs is the same as before (Figure 3) except for the mission time.

It should be noted here that the drag coefficients are supposed to be negligible when the controller and the trajectory planner are designed. Thus, when the controller and the trajectory planning are applied to the model (1), some of the thrusts exceed a little bit the safety margin due to model uncertainties. One can choose here to decrease \( \rho \) to less than 0.1 to obtain a \( t_f < 18.51 \) s. Since it is not always easy to manually tune \( \rho \), the adaptive trajectory planning is a better solution to deal with this issue.

C. Adaptive Trajectory Planning

To illustrate the adaptive trajectory planning, we use the first representation where \( N = 100 \) and \( \rho \) is initialized to 0.1. At \( t = 0 \), the trajectory planning is applied to the quad-rotor with \( \rho = 0.1 \). This parameter is then updated at each step time as explained in Section V (see Equation 48). Figure 6 shows the time evolution of \( \rho(t) \) during the mission. The only mismatch in our simulation is the drag coefficients that were neglected when the controller and the trajectory planner are designed. Thus, \( \rho \) is small and does not exceed the value of 0.05.
Applied normalized thrusts

Figure 5. Applied normalized thrust $T_i$ ($i = 1, \cdots, 4$) with trajectory planning

Time evolution of $\rho(t)$

Figure 6. Time evolution of $\rho(t)$

In our simulation, we have chosen to adapt the trajectory planning every 5 s until the condition $\|E(t)\| \leq 0.1 \|E(t_0)\|$ is verified. $E(t) = [e_1(t) \ e_2(t) \ e_3(t) \ e_4(t)]^T$ is the vector of tracking errors and $\|\cdot\|$ is the Euclidean norm. Figure 7 shows the time of the mission $t_f$ solution of the optimization problem. At the beginning of the mission and with $\rho = 0.01$, $t_f$ is equal to 18.51 s. Two trajectory adaptations are performed, the first at 5 s and the second at 10 s. At the first one, $t_f$ passes to 17.47 s and at the second one it passes to 16.59 s. Thus, adaptive planning allows gaining about 2 seconds with respect to non-adaptive planning.

Mission time $t_f$

Figure 7. Mission time $t_f$

The applied thrusts for the adaptive trajectory planning are illustrated in Figure 8. It can be seen that after the first adaptation at $t = 5$ s and with $\rho$ smaller than 0.1, the time of the mission $t_f$ is smaller and the allowable thrusts are larger. Thus, the adaptive trajectory planning allows a smarter planning in increasing
or decreasing $\rho$ during the mission in function of model uncertainties. The time evolutions of the flat outputs are similar to those given in Figure 3 except for the mission time $t_f$.

![Figure 8. Applied normalized thrust $T^*_i$ ($i = 1, \cdots, 4$) with adaptive trajectory planning](image)

VII. Conclusion

This paper considers the problem of control and trajectory planning for the non-linear 6 degrees of freedom quad-rotor UAV. It presents also some preliminary results on adaptive trajectory planning where the objective is to tune on-line the parameters of the optimization problem in function of model uncertainties. It should be noted here that the normalized thrust to be applied by each rotor to keep the system in hovering is equal to $T^*_i = mg/4T_{\text{max}} = 0.6125$ (see for example Figure 5). Thus, mathematically, the trajectory planning optimization problem admits a solution for $0 \leq \rho < 0.3875$ when the first representation is employed. The adaptive planning fails then if the model uncertainties are huge such that $\rho \geq 0.3875$. For such huge model uncertainties, the controller may even fail to keep the stability of the system and thus an additional effort should be paid for system modeling and/or robust control techniques. When the second representation is employed, the optimization problem admits a solution for $0 \leq \rho < 0.55$.

Since the adaptive trajectory planning consists in solving the optimization problem on-line, it is crucial to converge to a global solution. In our simulation, a multi-start optimization is performed to ensure the convergence of the problem. Future works will focus on applying the proposed approach to the quad-rotor UAV testbed at the Department of Mechanical and Industrial Engineering of Concordia University where the different calculations should be performed in real time.

Acknowledgments

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 and the Defence Research and Development Canada (DRDC) Technology Investment Fund.

References


