2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Decentralized Linear Time-Varying Model Predictive Control of a Formation of Unmanned Aerial Vehicles

Alberto Bemporad and Claudio Rocchi

Abstract— **This paper proposes a hierarchical MPC approach to stabilization and autonomous navigation of a formation of unmanned aerial vehicles (UAVs), under constraints on motor thrusts, angles and positions, and under collision avoidance constraints. Each vehicle is of quadcopter type and is stabilized by a local linear time-invariant (LTI) MPC controller at the lower level of the control hierarchy around commanded desired set-points. These are generated at the higher level and at a slower sampling rate by a linear time-varying (LTV) MPC controller per vehicle, based on an a simplified dynamical model of the stabilized UAV and a novel algorithm for convex under-approximation of the feasible space. Formation flying is obtained by running the above decentralized scheme in accordance with a leader-follower approach. The performance of the hierarchical control scheme is assessed through simulations, and compared to previous work in which a hybrid MPC scheme is used for planning paths on-line.**

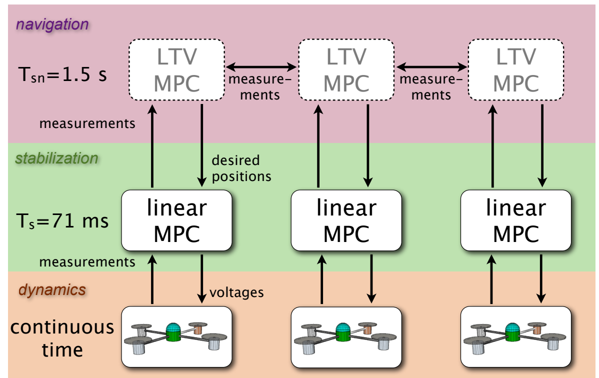

Fig. 1. Hierarchical control structure for UAV navigation

I. Introduction

The last few years have been characterized by an increasing interest in stabilizing and maneuvering a formation of multiple aerial vehicles. Research areas include both military and civilian applications (such as intelligence, reconnaissance, surveillance, exploration of dangerous environments) where Unmanned Aerial Vehicles (UAVs) can replace humans. VTOL (Vertical Take-Off and Landing) UAVs pose control challenges because of their highly nonlinear and coupled dynamics, and of limitations on actuators and pitch/roll angles. In particular, *quadcopters* are a class of VTOL vehicles for whose stabilization several approaches were proposed in literature, such as classical PID [1], nonlinear control [2], H_∞ control [3], and recently linear MPC (Model Predictive Control) [4].

MPC is particularly suitable for control of multivariable systems governed by costrained dynamics, as it allows one to operate closer to the boundaries imposed by hard constraints. In the context of UAVs, MPC techniques have been already applied for control of formation flight in [5]–[9] and for spacecraft rendezvous [10]–[12].

In the context of path planning for obstacle avoidance, several other solutions have been proposed in the literature, such as potential fields [13], [14], A^* with visibility graphs [1], nonlinear trajectory generation (see e.g. the NTG software package developed at Caltech [5]), vertex-graph (VGRAPH) algorithms [15], and mixed-integer linear programming (MILP) [16], [17].

A. Bemporad is with the IMT (Institutions, Market, Technologies) Institute for Advanced Studies Lucca, Italy, alberto.bemporad@imtlucca.it. C. Rocchi is with the Department of Mechanical and Structural Engineering, University of Trento, Italy, claudio.rocchi@ing.unitn.it. This work was partially supported by the European Space Agency through project "ROBMPC – Robust Model Predictive Control for Space Constrained Systems".

This paper adopts the two-layer MPC approach depicted in Figure 1 to stabilization and on-line trajectory generation for autonomous navigation with obstacle avoidance of a formation of quadcopters. At the lower level, linear constrained MPC controllers with integral action take care of stabilizing the quadcopters with offset-free tracking of desired set-points. At the higher hierarchical level and at a lower sampling rate, linear time-varying (LTV) MPC controllers generate on line the paths to follow for the formation to reach a given target position and shape while avoiding obstacles. The performance of the strategy adopted in this paper is assessed through simulations and compared to an alternative technique based on decentralized hybrid MPC proposed in a previous work [18]. The lower-level linear MPC algorithms control motor speeds directly via pulse position modulation, under admissible thrust and angle/position constraints. Based on a leader-follower approach in which the leader points to the target and the followers track a given relative position from the leader, the higher-level LTV-MPCs (one per vehicle) maintain the vehicles in formation towards a desired target in a decentralized way. Obstacles and vehicle-to-vehicle collisions are constraining predicted vehicle positions within a convex polyhedron contained in the feasible space, generated on-line via a novel approach proposed in this paper. We assume that target and obstacle positions may be time-varying and only known at run time, a situation for which off-line (optimal) planning cannot be easily accomplished.

The paper is organized as follows. After introducing the UAV dynamics in Section II, a linear MPC design is proposed in Section III-A for stabilization under constraints and trajectory tracking. Section III-B proposes a convex polyhedral approximation approach for obstacle avoidance, which is used in Section III-C to formulate the higher-level LTV-MPC controller for safe path planning. Section IV

978-1-61284-799-3/11/$26.00 ©2011 IEEE

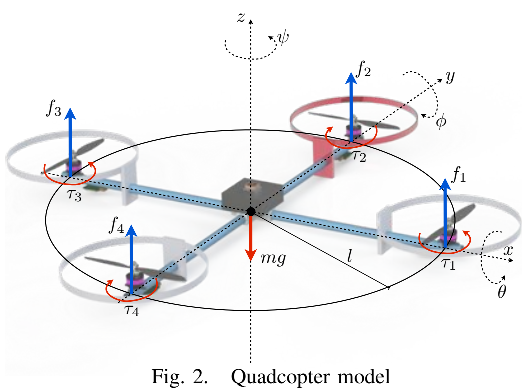

Fig. 2. Quadcopter model

provides simulation results and comparisons to the hybrid MPC approach described in [18]. Finally, some conclusions are drawn in Section V.

II. Nonlinear quadcopter dynamics

An aerial vehicle of quadcopter type is an underactuated mechanical system with six degrees of freedom and only four control inputs (see Figure 2). We denote by x, y, z the position of the vehicle and by θ, ϕ, ψ its rotations around the Cartesian axes, relative to the "world" frame. In particular, x and y are the coordinates in the horizontal plane, z is the vertical position, ψ is the yaw angle (rotation around the z-axis), θ is the pitch angle (rotation around the x-axis), and ϕ is the roll angle (rotation around the y-axis). The dynamical model adopted in this paper is mainly based on the model proposed in [19], simplified to reduce the computational complexity and to ease the design of the controller. As described in Figure 2, each of the four motors generate, respectively, four thrust forces f_1, f_2, f_3, f_4, and four torques τ_1, τ_2, τ_3, τ_4, which are adjusted by manipulating motor speeds. The resulting total force F and torques τ_θ, τ_ϕ, τ_ψ allow the change of the position and orientation coordinates of the quadcopter freely in the three-dimensional space:

$$\begin{aligned} &F = f_1 + f_2 + f_3 + f_4, \qquad && \tau_\theta = (f_2 - f_4)l \\ &\tau_\phi = (f_3 - f_1)l, && \tau_\psi = \sum_{i=1}^{4} \tau_i \end{aligned} \tag{1}$$

where l is the distance between each motor and the center of gravity of the vehicle.
Denote by $\tau = [\tau_\theta\ \tau_\phi\ \tau_\psi]'$ the torque vector, by $\eta = [\theta\ \phi\ \psi]'$ the angular coordinates vector, and by I the inertia matrix for the full-rotational kinetic energy of the UAV expressed directly in terms of the generalized coordinates η. The rotational dynamics of the quadcopter are expressed as

$$\tau = I\ddot{\eta} + \dot{I}\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial \eta}(\dot{\eta}' I \dot{\eta}) \tag{2}$$

As suggested in [2], we make the following simplification

$$\ddot{\eta} = \tilde{\tau} \tag{3}$$

where $\tilde{\tau} = [\tilde{\tau}_\theta\ \tilde{\tau}_\phi\ \tilde{\tau}_\psi]'$ is a new vector of control inputs. Through rotational transformations between the world frame and the quadcopter's body frame (placed on its center of gravity) we obtain the dynamical model

$$\begin{aligned} m\ddot{x} &= -F \sin\theta - \beta\dot{x} \\ m\ddot{y} &= F \cos\theta \sin\phi - \beta\dot{y} \\ m\ddot{z} &= F \cos\theta \cos\phi - mg - \beta\dot{z} \\ \ddot{\theta} &= \tilde{\tau}_\theta,\ \ \ddot{\phi} = \tilde{\tau}_\phi,\ \ \ddot{\psi} = \tilde{\tau}_\psi \end{aligned} \tag{4}$$

where m is the mass of the UAV, and the damping factor β takes into account friction effects that affect the real vehicle.

Trying to adjust directly the torques is not a practical approach. Therefore, as in [18], at the price of a slight increase in model complexity the following relations are used:

$$\begin{aligned} \ddot{x} &= (-u_1 \sin\theta - \beta\dot{x})\frac{1}{m} \\ \ddot{y} &= (u_1 \cos\theta \sin\phi - \beta\dot{y})\frac{1}{m} \\ \ddot{z} &= -g + (u_1 \cos\theta \cos\phi - \beta\dot{z})\frac{1}{m} \\ \ddot{\theta} &= \frac{u_2}{I_{xx}},\ \ \ddot{\phi} = \frac{u_3}{I_{yy}},\ \ \ddot{\psi} = \frac{u_4}{I_{zz}} \end{aligned} \tag{5}$$

in which

$$\begin{aligned} &u_1 = f_1 + f_2 + f_3 + f_4,\ \ u_2 = (f_2 - f_4)l \\ &u_3 = (f_3 - f_1)l,\ \ u_4 = (-f_1 + f_2 - f_3 + f_4)l \end{aligned} \tag{6}$$

g is the gravity acceleration, and I_{xx}, I_{yy}, I_{zz} are the components of diagonal inertia matrix of the airframe at its center of mass. The parameters used for the quadcopter in this work are reported in Table I.

TABLE I

Quadcopter parameters

m [kg]	l [m]	β [Ns/m]	I_{xx} [Nms^2]	I_{yy} [Nms^2]	I_{zz} [Nms^2]
1.846	0.505	0.2	0.1722	0.1722	0.3424

When using brushless motors, continuous voltage control is replaced by *Electronic Speed Controller* (ESC), that adjusts motor speed by *Pulse Position Modulation* (PPM), as it is standard in RC plane technology. Briefly, PPM is a *phase modulation*: The speed of the motor (or the servo angle) is regulated by the *position* of an impulse of fixed amplitude and length within the control signal period. So an ESC gets that position μ_m, expressed in microseconds (μs) as the input value. Standard ESCs input values can vary in the 1000μs - 2000μs range. Higher velocities (angles) correspond to higher input values.
All motors are supposed to share the same technical specs and response, fast enough to neglect actuation delays; they have a nonlinear behavior that can be approximated by a piecewise affine function consisting of three affine terms, $p_j\mu_{mi} - q_j$, $i = 1, \ldots, 4$, $j = 1, 2, 3$. Therefore, each motor thrust f_i is modeled as a piecewise linear function of the applied value μ_{mi}:

$$f_i = \frac{g(p_j\mu_{mi} - q_j)}{1000}, \quad i = 1, \ldots, 4, \quad j = 1, 2, 3. \tag{7}$$

In summary, the quadcopter is controlled by regulating microsecond values according to (6)–(7). The obtained nonlinear dynamical model has twelve states (six positions and six velocities) and four inputs (the motors microseconds

μ_{mi}), largely coupled through the nonlinear relations (5). The nonlinear model (5)-(7) will be used to simulate closed-loop trajectories in Section IV.

III. Hierarchical MPC of each UAV

Consider the hierarchical control system depicted in Figure 1. At the top layer, LTV-MPC controllers generate on-line the desired positions (x_d, y_d, z_d) to the lower stabilization layer, in order to accomplish the main mission, namely reach a given target position (x_t, y_t, z_t) while avoiding collisions with possible obstacles and other UAVs. The desired positions are tracked in real-time by linear MPC controllers placed at the middle layer of the architecture (these might be as well replaced by linear controllers). The bottom layer is the physical layer described by the nonlinear dynamics of the quadcopter, whose motor speeds are commanded by the linear MPC controllers. In the next sections we describe in details each layer of the proposed architecture.

A. Linear MPC for stabilization

In order to design a linear MPC controller to stabilize the quadcopter vehicle on given desired positions/angles, we linearize the nonlinear dynamical model (5) around an equilibrium condition of hovering and approximate the motor characteristics via a single linear function. The resulting linear continuous-time state-space system is converted to discrete-time with sampling time T_s

$$\begin{cases} \xi_L(k+1) &= A\xi_L(k) + Bu_L(k) \\ y_L(k) &= \xi_L(k) \end{cases} \tag{8}$$

where $\xi_L(k) = [\theta, \phi, \psi, x, y, z, \dot{\theta}, \dot{\phi}, \dot{\psi}, \dot{x}, \dot{y}, \dot{z}]' \in \mathbb{R}^{12}$ is the state vector, $u_L(k) = [\mu_{m1}, \mu_{m2}, \mu_{m3}, \mu_{m4}]' \in \mathbb{R}^4$ is the input vector, $y_L(k) \in \mathbb{R}^{12}$ is the output vector (that we assume completely measured or estimated), and A, B, C, D are matrices of suitable dimensions obtained by the linearization process. The linear MPC formulation of the Model Predictive Control Toolbox for MATLAB [20] based on quadratic programming is used to design the stabilizing controller under the stated input and output constraints.

B. Convex approximations for obstacle avoidance

Let $p = [x\ y\ z]'$ denote the position of the vehicle and let M denote the number of obstacles to be avoided. Each obstacle is described by a convex polyhedron $W_i \subset \mathbb{R}^3$ centered on a different point $q_i \in \mathbb{R}^3$, that is the set $\{q_i\} \oplus W_i$ is considered as infeasible, $i = 1, \ldots, M$. Non-convex obstacles can be modeled by overlapping several of such convex shapes.

In order to impose collision avoidance constraints as (possibly time-varying) linear constraints, the nonconvex feasible space where the vehicle can navigate must be under-approximated by a convex polyhedron. A novel fast greedy algorithm to maximize the size of a polyhedron not containing a set of points is described in the sequel for a generic space-dimension d.

Lemma 1: Let $p_0, q_1, q_2, \ldots, q_M \in \mathbb{R}^d$, with $p_0 \neq q_i$, $\forall i = 1, \ldots, M$. The polyhedron $P = \{p \in \mathbb{R}^d :\ A_c p \leq b_c\}$ with

$$A_c = \begin{bmatrix} (q_1 - p_0)' \\ \vdots \\ (q_M - p_0)' \end{bmatrix},\ b_c = \begin{bmatrix} (q_1 - p_0)' q_1 \\ \vdots \\ (q_M - p_0)' q_N \end{bmatrix} \tag{9}$$

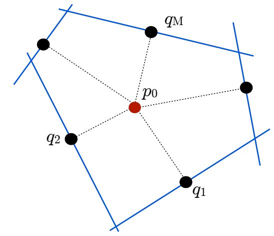

Fig. 3. Convex polyhedron around the current vehicle position p avoiding point obstacles $q_1, \ldots, q_M$

contains p_0 in its interior and does not contain any of the points q_i in its interior, for all $i = 1, \ldots, M$.

Proof: As sketched in Figure 3, the halfspace

$$H_i = \{p \in \mathbb{R}^d :\ (q_i - p_0)' p \leq (q_i - p_0)' q_i\} \tag{10}$$

contains q_i on its boundary $\partial H_i = \{p \in \mathbb{R}^d :\ (q_i - p_0)' p = (q_i - p_0)' q_i\}$, and is such that ∂H_i is orthogonal to $q_i - p_0$. Moreover $(q_i - p_0)' p_0 = (q_i - p_0)'(p_0 - q_i + q_i) = -\|q_i - p_0\|^2 + (q_i - p_0)' q_i < (q_i - p_0)' q_i$ implies that p_0 is in the interior $\mathring{H}_i = H_i \setminus \partial H_i = \{p \in \mathbb{R}^d :\ (q_i - p_0)' p < (q_i - p_0)' q_i\}$ of H_i, $\forall i = 1, \ldots, M$. Since $P = \cap_{i=1}^M H_i$, it follows that $p \in \mathring{H}_i$. Assume by contradiction that there exists a point $q_i \in \mathring{P}$. Then there is a scalar $\sigma > 0$ such that $q_\sigma = q_i + \sigma(q_i - p_0) \in P$, a contradiction since $(q_i - p_0)' q_\sigma = (q_i - p_0)' q_i + \sigma \|q_i - p_0\|^2 > (q_i - p_0)' q_i$ violates an inequality defining P. $\square$

Note that some of the halfspaces H_i in (10) may be redundant, so (A, b) in (9) may not be a minimal hyperplane representation of P.

Lemma 2: Let $p_0, q_1, q_2, \ldots, q_M \in \mathbb{R}^d$, with $p_0 \neq q_i$, $\forall i = 1, \ldots, M$, and let $W_1, \ldots, W_M$ be polyhedra in $\mathbb{R}^d$. Let A_c, b_c be defined as in (9) and let $g \in \mathbb{R}^M$ such that its j-th component g^j defined as

$$\begin{array}{ll} g^j = \min_{w \in \mathbb{R}^d} & A_c^j w \\ \text{s.t.} & w \in W_j \end{array} \tag{11}$$

for $j = 1, \ldots, M$. Then the polyhedron $P = \{p \in \mathbb{R}^d : A_c p \leq b_c + g\}$ does not contain any polyhedron $B_j = \{q_j\} \oplus W_j$ in its interior, $\forall j = 1, \ldots, M$.

Proof: Assume by contradiction that there exists a point $p \in \mathring{P} \cap B_j$, that is $A_c p < b_c + g$, $p \in B_j$. The latter condition implies that $p = q_j + w$ for some $w \in W_j$, and hence $b_c^j + g^j > A_c^j p = A_c^j(q_j + w) = A_c^j q_j + A_c^j w \geq A_c^j q_j + g^j$, which implies $A_c^j q_j < b_c^j$. By Lemma 1 this is a contradiction. $\square$

Note that if W_j's are polytopes and their vertex representation $W_j = \text{conv}\{w_{j1}, \ldots, w_{js_j}\}$ is available, then (11) can be simply solved as

$$g^j = \min_{h=1,\ldots,s_j} A_c^j w_{jh} \tag{12}$$

for $j = 1, \ldots, M$. Moreover, for any given scaling $\mu_j W_j = \text{conv}\{\mu_j w_{j1}, \ldots, \mu_j w_{js_j}\}$ of W_j, $\mu_j \geq 0$, the corresponding component g^j in (12) simply scales as $\mu_j g^j$.

Figure 4 shows a two-dimensional example including six obstacles for subsequent positions $p_0 = p(0)$, $p_0 = p(1)$, ..., $p_0 = p(5)$, representing an ideal vehicle moving towards a target x_t. The main idea of the proposed approach is

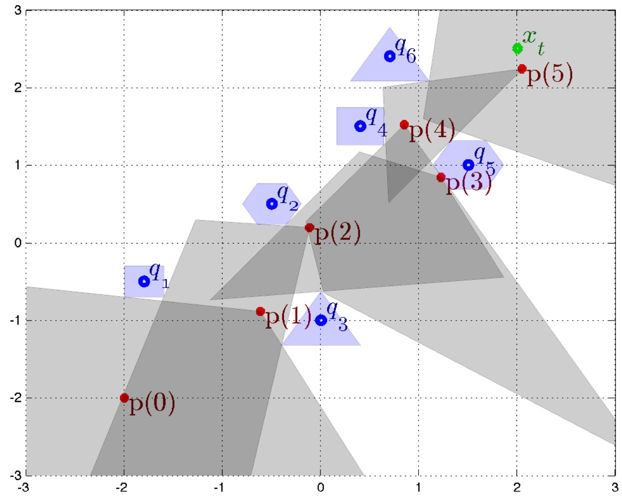

Fig. 4. Example of feasible polyhedra for navigation among obstacles. The subsequent position $p(j+1)$ minimizes the Euclidean distance from x_t within the feasible polyhedron around $p(j)$

that, by taking into account multiple time steps, the union of all pointwise-in-time convex approximations provides a rather good non-convex approximation of the feasible space of interest for navigation.

To take into account that the size of the UAV is not negligible compared to the distance between the obstacles and the vehicle, introduce vectors $d_h \in \mathbb{R}^3$, $h = 1, \ldots, r$ and let the vehicle be contained in the polyhedron $\{p\} + \mathrm{conv}(d_1, \ldots, d_r)$ ($r = 1$ and $d_1 = 0$ in case the size of the UAV is considered negligible). Based on the previous lemmas, the following theorem is immediate to prove.

Theorem 1: Let $\mathcal{V} \triangleq \mathrm{conv}(p_0 + d_1, \ldots, p_0 + d_r)$, $p_0, d_1, \ldots, d_r \in \mathbb{R}^d$, $r \geq 1$, let $q_1, q_2, \ldots, q_M \in \mathbb{R}^d$, with $p_0 \neq q_i$, $\forall i = 1, \ldots, M$, and let $W_1, \ldots, W_M$ be polyhedra in $\mathbb{R}^d$. Let A_c, b_c be defined as in (9) and let $g \in \mathbb{R}^M$ defined as in (11). Let $P_\mathcal{V} = \{p \in \mathbb{R}^d : \; A_c p \leq b_c + g - A_c d_i, \; i = 1, \ldots, r\}$ and assume that $P_\mathcal{V}$ is nonempty. Then $\mathcal{V} \subseteq P_\mathcal{V}$ and $\mathring{P}_\mathcal{V} \cap B_j = \emptyset$, $\forall j = 1, \ldots, M$, where $B_j = \{q_j\} \oplus W_j$.

C. LTV-MPC navigation algorithm

The closed-loop dynamics composed by the quadcopter and the linear MPC controller can be approximated as a first order system. The dynamical model of vehicle i is described by

$$p_i(t+1) = A_i p_i(t) + B_i p_{ci}(t) \tag{13}$$

where $p_{ci}(t) = [x_{ci}(t) \;\; y_{ci}(t) \;\; z_{ci}(t)]'$ is the position commanded at time t, $p_i(t) = [x_i(t) \;\; y_i(t) \;\; z_i(t)]'$ is the actual current position of the vehicle, $A_i = \mathrm{diag}(e^{-T_{\mathrm{sn}}/\tau_{xi}}, e^{-T_{\mathrm{sn}}/\tau_{yi}}, e^{-T_{\mathrm{sn}}/\tau_{zi}})$, $B_i = I - A_i$, and T_{sn} is the sampling time. T_{sn} is chosen large enough to neglect fast transient dynamics, so that the lower and upper MPC designs can be conveniently decoupled. Let $\{p(t)\} + \mathrm{conv}(d_1(t), \ldots, d_r(t))$ be a polyhedron containing the vehicle at time t, $r \geq 1$, $d_1(t), \ldots, d_r(t) \in \mathbb{R}^3$, $\forall t \geq 0$ (time-varying values for vectors $d_i(t)$ can be useful to take rotations into account and if the vehicle is not treated as a rigid body).

The linear constraints associated with matrices A_c, b_c and g, obtained by (9) and (12), respectively, to avoid obstacles, depend on the current position $p(t) \in \mathbb{R}^3$ and in general vary as time evolves. Therefore we adopt the following LTV-MPC algorithm. At time t, the following problem is solved via quadratic programming

$$\begin{aligned}
\min \quad & \rho\epsilon^2 + \sum_{k=0}^{N-1} \|w_y(p(t+k) - p_d)\|^2 + \\
& \|w_{\Delta u}(p_c(t+k) - p_c(t+k-1))\|^2 \\
\text{s.t.} \quad & p_i(t+k+1) = A_i p_i(t+k) + B_i p_{ci}(t+k) \\
& \qquad k = 0, \ldots, N-1 \\
& \Delta_{\min} \leq p_c(t+k) - p_c(t+k-1) \leq \Delta_{\max} \\
& \qquad k = 0, \ldots, N_u - 1 \\
& p_c(t+k) = p_c(t+N_u-1), \;\; k = N_u, \ldots, N \\
& A_c(t+k)p(t+k) \leq b_c(t+k) + g(t+k) - A_c(t+k) \cdot \\
& \cdot d_h(t+k) + \mathbb{1}\epsilon, \;\; k = 0, \ldots, N-1, \;\; h = 1, \ldots, r
\end{aligned} \tag{14}$$

The scalars $w_y, w_{\Delta u} \in \mathbb{R}$, $w_y, w_{\Delta u} > 0$ are the weights on outputs and inputs, respectively. Matrices $A_c(t+k)$, $b_c(t+k)$, $g(t+k)$ are obtained by (9) and (11) by setting $p_0 = p(t)$ (current vehicle position), $q_i = q_i(t+k)$ (predicted obstacle position, where $q_i(t+k) \equiv q_i(t)$ when obstacles are considered fixed in prediction), $W = \mathrm{conv}\{w_1(t+k), \ldots, w_s(t+k)\}$ and $\mu = \mu(t+k)$ (predicted size and magnitude of obstacles, respectively, where $w_i(t+k) \equiv w_i(0)$ and $\mu = \mu(0)$ when the obstacle shape and size is assumed time-invariant). Moreover, as it is standard practice in all practical MPC implementations, the slack variable ϵ is used to soften the obstacle avoidance constraints, therefore avoiding that (14) is infeasible, and is penalized by a large weight $\rho > 0$ in (14).

In summary, the proposed hierarchical MPC approach for stabilization and navigation of each UAV consists of the following steps: (i) the LTV-MPC control law chooses the optimal desired position $p_c(t)$ every T_{sn} time units by solving problem (14) that makes the vehicle position $p(t)$ approaching the target position p_d while avoiding collisions; (ii) the LTI-MPC control law is executed at a shorter sampling time T_s to stabilize the vehicle around $p_c(t)$ under actuator and angle/position constraints.

D. Formation flying

The hierarchical MPC structure described above for one vehicle is extended to coordinate a formation of V cooperating UAVs, $V > 1$. We use a decentralized leader-follower approach to manage the formation, where one of the vehicles (the leader) tracks a desired target position p_t and all the other vehicles (the followers) track a desired constant relative distance p_{di} from the leader. Each vehicle treats the other UAVs in the formation as further obstacles to avoid, so that the total number of obstacles accounts now for both the real ones and the other vehicles.

Each UAV is equipped with its own MPC control hierarchy and takes decisions autonomously, measuring its own state and the positions of the other vehicles and obstacles. The formation must be capable of reconfiguring, making decisions (for instance, changing relative distances to modify the formation shape), and achieving mission goals (e.g., target tracking). To take into account the nonzero dimensions of the UAVs, these are modeled as constant parallelepipeds $\text{conv}(d_1, \ldots, d_r) = \text{conv}(w_1, \ldots, w_s)$, $r = s = 8$, whose height (defined along the z-axis) is half their width and depth.

To improve cooperativeness of the UAVs, at the current time t vehicle $\#i$, besides knowing the position $q_j(t)$ of the other vehicles, $j = 1, \ldots, V$, $j \neq i$, each follower is aware of the previous optimal sequence computed by the other vehicles $p_c^j(t|t-1), p_c^j(t+1|t-1), \ldots, p_c^j(t+N-2|t-1)$. Such sequences are used to predict the future obstacle positions $q_j(t+k)$ via the dynamical model (13) in which j replaces i, under the assumption $p_c^j(t+N-1|t-1) = p_c^j(t+N-2|t-1)$. This provides a better estimate of the free polyhedral space for collision avoidance.

IV. Simulation results

We test the proposed decentralized and hierarchical MPC scheme for navigation of a formation of three quadcopters moving to a target point in the presence of four obstacles (tetrahedra) to avoid.

The linear MPC controller for constrained stabilization is designed by using the standard setup of the Model Predictive Control Toolbox for MATLAB [21], along with the following specifications: input constraints 1130 μs$\leq \mu_{mj} \leq$1540 μs and constant weight $w^{\Delta u} = 0.1$ on each input increment $\Delta\mu_{mj}$, $j = 1, 2, 3, 4$, output constraint $z \geq 0$ m on vehicle altitude and $-\frac{\pi}{6} \leq \theta, \phi \leq \frac{\pi}{6}$ on pitch and roll angles, output weights $w_y = 0$ on θ and ϕ tracking errors, and $w_y = 10$ on all remaining tracking errors. The chosen set of weights ensures a good trade-off between fast system response and energy spent for actuation. The prediction horizon is $N_L = 20$, the control horizon is $N_{Lu} = 3$, which, together with the choice of weights, allow obtaining a good compromise between tracking performance, robustness, and computational complexity. The sampling time of the controller is $T_s = \frac{1}{14}$ s. The remaining parameters of the MPC controller are defaulted by the Model Predictive Control Toolbox for MATLAB.

For LTV-MPC control, the MPCSofT Toolbox for MATLAB is adopted, that has been developed at the University of Trento within the activities of the European Space Agency project "ROBMPC". The MPCSofT Toolbox allows one to specify rather arbitrary LTV prediction models and constraints in Embedded MATLAB and to efficiently set up and solve the quadratic programming problem associated with the MPC problem in real-time.

The following parameters are employed for model (13): τ_{xi}=2.82 s τ_{yi}=2.85 s, τ_{zi}=2.12 s, for all UAVs, $i = 1, 2, 3$. For the LTV-MPC setup we set prediction horizon $N = 10$, control horizon $N_u = 5$, sampling time T_{sn}=1.5 s, weights $w_y = 0.1$ on all outputs, $w_{\Delta u} = 0.1$ on all input increments, and $\Delta_{\max} = -\Delta_{\min} = 0.5$ m as the maximum rate of change of the desired position p_c of each vehicle. The obstacles are modeled as tetrahedra

$$W = \text{conv}(\begin{bmatrix} -1/3 \\ -1/3 \\ -1/2 \end{bmatrix}, \begin{bmatrix} 2/3 \\ -1/3 \\ -1/2 \end{bmatrix}, \begin{bmatrix} -1/3 \\ 2/3 \\ -1/2 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1/2 \end{bmatrix})$$

the scaling factor is $\mu = 7$, and ρ_1=1000 is used to weight the slack variable ϵ used for soft constraints on obstacle avoidance. The initial positions of the UAVs are $p_L(0) \triangleq [x_L(0)\ y_L(0)\ z_L(0)]' = [6\ 6\ 0]'$ for the leader (all quantities are expressed in MKS international units), $p_{F1}(0) \triangleq [x_{F1}(0)\ y_{F1}(0)\ z_{F1}(0)]' = [4\ 4\ 0]'$, and $p_{F2}(0) \triangleq [x_{F2}(0)\ y_{F2}(0)\ z_{F2}(0)]' = [2\ 2\ 0]'$ for the followers; the target point for the leader is located at $p_t = [\bar{x}_t\ \bar{y}_t\ \bar{z}_t] = [40\ 40\ 6]'$; the followers take off with a delay of 2.5 and 5 seconds respectively, and should follow the leader at given distances $p_L - p_{d1}, p_{d1} = [6\ 1\ 0]'$ and $p_L - p_{d2}, p_{d2} = [1\ 6\ 0]'$, respectively. The results were obtained on a Core 2 Duo running MATLAB R2009b, the Model Predictive Control Toolbox for MATLAB, and the MPCSofT Toolbox under MS Windows. The trajectories obtained by using the proposed decentralized hierarchical LTV-MPC + LTI-MPC approach are shown in Figure 5. The performance is quite satisfactory: The trajectories circumvent obstacles without collisions and finally the quadcopters settle at the target points, while maintaining the desired formation as much as the obstacles allow for keeping it. On average the LTV-MPC action for set-point generation requires about 75 ms per sample step (T_{sn}=1.5 s), the linear MPC control action about 150 μs per sample step (T_s=1/14 s).

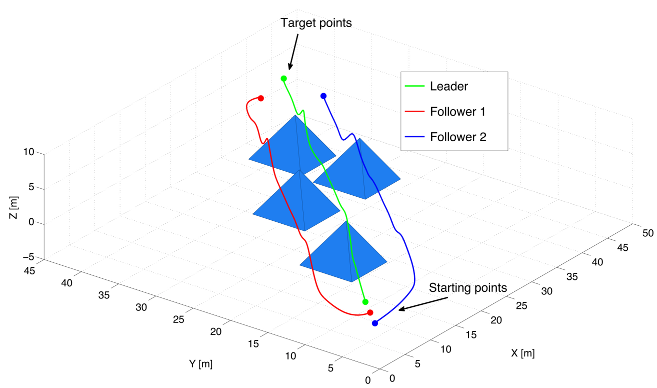

Fig. 5. Trajectories of formation flying under obstacle avoidance constraints, LTV-MPC approach

A. Comparison with decentralized hybrid MPC

A decentralized hybrid MPC approach for solving the same problem tackled in this paper was proposed in [18]. In order to compare the two strategies we use the same simulation scenario described in the previous section. The resulting performance figures of the two approaches are similar, as shown in Figure 6. The hybrid MPC approach takes an average CPU time of 45 ms per time step (T_{sn}=1.5 s) to compute the control action using the commercial and highly optimized MIQP solver IBM CPLEX [22]. Note that while the CPU time for the navigation algorithm is similar in both approaches, the complexity of the hybrid MPC code is much higher, being based on the CPLEX library. On the other hand, the LTV-MPC code is immediately deployable by using the C-code generation functionality of the MATLAB environment.

Finally, a quantitative comparison of the two different control strategies is reported in Table II. The following

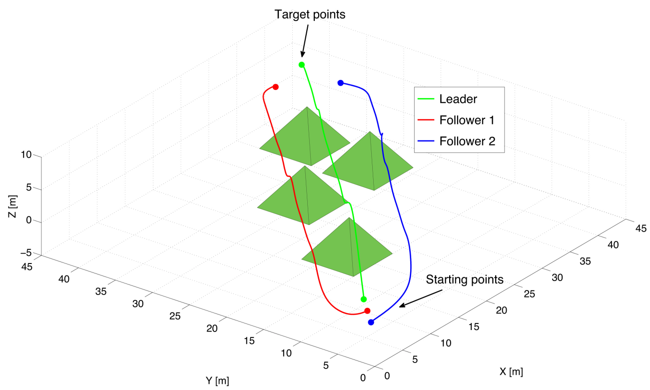

Fig. 6. Trajectories of formation flying under obstacle avoidance constraints, hybrid MPC approach

three performance indices defined on the simulation interval 25÷220 s (i.e., 350÷3080 samples) are considered:

$$J_{\rm tt} = \sum_{k=350}^{3080} \|p_L(k) - p_t\|_2^2$$
$$J_{\rm fpt} = \sum_{k=350}^{3080} \|p_L(k) - p_{F1}(k) - p_{d1}\|_2^2 + \|p_L(k) - p_{F2}(k) - p_{d2}\|_2^2$$
$$J_{\rm u} = \sum_{k=350}^{3080} \|u(k) - u(k-1)\|_1$$

where $J_{\rm tt}$ represents the *target tracking* Integral Square Error (ISE) index, $J_{\rm fpt}$ the *formation pattern tracking* ISE index, and $J_{\rm u}$ the *absolute derivative of input signals* (IADU) index for checking the smoothness of control signals [3]. The indices are normalized with respect to the values obtained using the decentralized hybrid MPC strategy. It is apparent that the two strategies have similar performance figures in target tracking and IADU (this is due to a similar setup of MPC weights and constraints), while in keeping the desired formation the LTV-MPC approach is superior to the hybrid MPC method.

TABLE II
COMPARISON OF LTV-MPC AND HYBRID MPC APPROACHES

	$J_{\rm tt}$	$J_{\rm fpt}$	$J_{\rm u}$
decentralized hybrid MPC	1	1	1
decentralized LTV MPC	-2.46%	-16.30%	+9.20%

V. CONCLUSIONS

This paper has proposed a decentralized hierarchical MPC approach to stabilization and navigation of a formation of UAVs under various constraints. In particular, despite the convex approximation of the obstacle-free space, an LTV-MPC approach was proved very effective in navigating under obstacle avoidance conditions, the conservativeness of the convex approximation being mitigated by its time-varying nature and by the receding-horizon approach of MPC. The obtained performance is comparable to that achievable by using more complex methods, such as hybrid MPC approaches. Compared to off-line planning methods, the proposed hierarchical MPC scheme generates the 3D path to follow completely on-line, which is particularly appealing in realistic scenarios where the positions of the target and of the obstacles, and the shapes of the latter, may not be known in advance, but rather acquired (and possibly time-varying) during flight operations.

REFERENCES

[1] G. Hoffmann, S. Waslander, and C. Tomlin, "Quadrotor helicopter trajectory tracking control," in *Proc. AIAA Guidance, Navigation, and Control Conf., Honolulu, HI*, 2008.
[2] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 510–516, 2004.
[3] G. Raffo, M. Ortega, and F. Rubio, "An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter," *Automatica*, vol. 46, pp. 29–39, 2010.
[4] A. Bemporad, C. Pascucci, and C. Rocchi, "Hierarchical and hybrid model predictive control of quadcopter air vehicles," in *3rd IFAC Conference on Analysis and Design of Hybrid Systems, Zaragoza, Spain*, 2009.
[5] W. Dunbar and R. Murray, "Model predictive control of coordinated multi-vehicle formations," in *IEEE Conference on Decision and Control*, vol. 4, 2002, pp. 4631–4636.
[6] F. Borrelli, T. Keviczky, K. Fregene, and G. Balas, "Decentralized receding horizon control of cooperative vechicle formations," in *Proc. 44th IEEE Conf. on Decision and Control and European Control Conf.*, Sevilla, Spain, 2005, pp. 3955–3960.
[7] W. Li and C. Cassandras, "Centralized and distributed cooperative receding horizon control of autonomous vehicle missions," *Mathematical and computer modelling*, vol. 43, no. 9-10, pp. 1208–1228, 2006.
[8] A. Richards and J. How, "Decentralized model predictive control of cooperating UAVs," in *Proc. 43rd IEEE Conf. on Decision and Control*, 2004, pp. 4286–4291.
[9] V. Manikonda, P. Arambel, M. Gopinathan, R. Mehra, F. Hadaegh, S. Inc, and M. Woburn, "A model predictive control-based approach for spacecraft formation keeping and attitude control," in *American Control Conference, 1999. Proceedings of the 1999*, vol. 6, 1999.
[10] A. Richards and J. How, "Performance evaluation of rendezvous using model predictive control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. Citeseer, 2003.
[11] E. Hartley, "Model predictive control for spacecraft rendezvous," Ph.D. dissertation, University of Cambridge, 2010.
[12] H. Park, S. Di Cairano, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking with a rotating/tumbling platform and for debris avoidance," in *2011 American Control Conference, San Francisco, California, USA*, June 29 - July 1, 2011.
[13] J. Chuang, "Potential-based modeling of three-dimensional workspace for obstacle avoidance," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 778–785, 1998.
[14] T. Paul, T. Krogstad, and J. Gravdahl, "Modelling of UAV formation flight using 3D potential field," *Simulation Modelling Practice and Theory*, vol. 16, no. 9, pp. 1453–1462, 2008.
[15] T. Lozano-Pérez and M. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, p. 570, 1979.
[16] A. Richards and J. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 3, 2002.
[17] F. Borrelli, T. Keviczky, G. Balas, G. Stewart, K. Fregene, and D. Godbole, "Hybrid decentralized control of large scale systems," ser. Lecture Notes in Computer Science, M. Morari and L. Thiele, Eds. Springer-Verlag, 2005, pp. 168–183.
[18] A. Bemporad and C. Rocchi, "Decentralized hybrid model predictive control of a formation of unmanned aerial vehicles," in *18th IFAC World Congress, Milan, Italy*, Aug. 28-Sep. 2, 2011 (accepted for publication), available at http://control.ing.unitn.it/aero.
[19] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," Master's thesis, Department of Automatic Control, Lund University, October 2008.
[20] A. Bemporad, N. Ricker, and J. Owen, "Model Predictive Control-New tools for design and evaluation," in *American Control Conference*, vol. 6, Boston, MA, 2004, pp. 5622–5627.
[21] A. Bemporad, M. Morari, and N. Ricker, *Model Predictive Control Toolbox for Matlab – User's Guide*. The Mathworks, Inc., 2004, http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/.
[22] ILOG, Inc., *CPLEX 11.2 User Manual*, Gentilly Cedex, France, 2008.