Decentralized Hybrid Model Predictive Control of a Formation of Unmanned Aerial Vehicles

A. Bemporad ∗ C. Rocchi ∗

Abstract: This paper proposes a hierarchical MPC strategy for autonomous navigation of a formation of unmanned aerial vehicles (UAVs) of quadcopter type under obstacle and collision avoidance constraints. Each vehicle is stabilized by a lower-level local linear MPC controller around a desired position, that is generated, at a slower sampling rate, by a hybrid MPC controller per vehicle. Such an upper control layer is based on a hybrid dynamical model of the UAV in closed-loop with its linear MPC controller and of its surrounding environment (i.e., the other UAVs and obstacles). The resulting decentralized scheme controls the formation based on a leader-follower approach. The performance of the hierarchical control scheme is assessed through simulations and comparisons with other path planning strategies, showing the ability of linear MPC to handle the strong couplings among the dynamical variables of each quadcopter under motor voltage and angle/position constraints, and the flexibility of the decentralized hybrid MPC scheme in planning the desired paths on-line.

Keywords: Model predictive control, hierarchical control, aerospace, decentralized control, hybrid systems, obstacle avoidance, unmanned aerial vehicles

1. INTRODUCTION

The last few years have been characterized by an increasing interest in stabilizing and maneuvering a formation of multiple 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. There are different types of UAVs: planes operating 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 Dunbar [2001], Dunbar and Murray [2002], Borrelli et al. [2005b], Li and Cassandras [2006], Richards and How [2004], Manikonda et al. [1999].

In the context of path planning for obstacle avoidance, several other solutions have been proposed in the literature, such as potential fields [Chuang, 1998, Paul et al., 2008], A* with visibility graphs [Hoffmann et al., 2008, Latombe, 1991], nonlinear trajectory generation (see e.g. the NTG software package developed at Caltech [Dunbar and Murray, 2002]), and mixed-integer linear programming (MILP) [Richards and How, 2002, Borrelli et al., 2005a]. In particular the latter has shown the great flexibility of on-line mixed-integer optimization in real-time trajectory planning of aircrafts, as also reported in Pallottino et al. [2002] where on-line MILP techniques were proved very effective in handling multi-aircraft conflict avoidance problems.

This paper extends the two-layer MPC approach to quadcopter stabilization and on-line trajectory generation for autonomous navigation with obstacle avoidance presented earlier in Bemporad et al. [2009]. A linear constrained MPC controller with integral action takes care of quadcopter stabilization and offset-free tracking of desired setpoints. At a higher hierarchical level and lower sampling rate, a hybrid MPC controller generates on-line the path to follow to reach a given target position/orientation while avoiding obstacles. The approach is extended in two ways: first, we let the underlying linear MPC algorithm controlling directly motor voltages, rather than torques; second, the higher-level hybrid MPC controllers are organized in
Fig. 1. Hierarchical control structure for UAV navigation

debb central control scheme, based on a leader-follow
approach. We assume that target and obstacle positions
may be time-varying, and in this case that only their cur-
rent coordinates are known (not the future ones), so that
off-line (optimal) planning cannot be easily accomplished.

The paper is organized as follows. In Section 2 we describe
the layers of the hierarchical structure of each controlled
quadcopter and its nonlinear dynamics, whose lineariza-
tion provides the prediction model for linear MPC de-
sign for stabilization under constraints and tracking of
generated trajectories. The higher-level hybrid MPC con-
troller for path planning and obstacle avoidance is also
introduced. Section 3 describes the potential fields method
described in Paul et al. [2008] and results are compared in
simulation in Section 4. Finally, some conclusions are
drawn in Section 5.

2. HIERARCHICAL MPC OF EACH UAV

Consider the hierarchical control system depicted in
Figure 1. At the top layer a hybrid MPC controller generates
on-line the desired positions \((x_d, y_d, z_d)\), in order to ac-
complish the main mission, namely reach a given target
position \((x_t, y_t, z_t)\) while avoiding collisions with obstacles
and other UAVs. Such references \((x_d, y_d, z_d)\) are tracked
in real-time by a linear MPC controller placed at a lower
layer of the architecture. The bottom layer depicted in Figure 1
is the physical layer representing the nonlinear dynamics
of the quadcopters. In the next subsections we describe in
details each layer of the architecture.

2.1 Nonlinear quadcopter dynamics

A quadcopter aerial vehicle is an underactuated me-
chanical 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 \(\theta, \phi, \psi\) its rotations around
the Cartesian axes, relative to the “world” frame \(I\). In
particular, \(x\) and \(y\) are the coordinates in the horizontal
plane, \(z\) is the vertical position, \(\psi\) is the yaw angle (ro-
atation around the \(z\)-axis), \(\theta\) is the pitch angle (rota-
ation around the \(x\)-axis), and \(\phi\) is the roll angle (rotation around the \(y\)-axis). The dynamical model adopted in this paper is
mainly based on the model proposed in Bresciani [2008],
simplified to reduce the computational complexity and to
ease control design. As described in Figure 2, each of the

\[ f_i = 9.81(22.5V_{M_i} - 9.7)/1000, \quad i = 1, \ldots, 4 \]

approximated to consider all motors identical, to express
the relationships between motors thrusts and their volt-
gages. A total force \(f\) and three torques \(\tau_x, \tau_y, \tau_z\) around
their corresponding axes are generated

\[
\begin{align}
    \tau_x &= f_1 + f_2 + f_3 + f_4 \\
    \tau_y &= (f_2 - f_4)l \\
    \tau_z &= (f_3 - f_1)l \\
    \tau_x &= \sum_{i=1}^{4} \tau_i 
\end{align}
\]

where \(l\) is the distance between each motor and the center
of gravity of the vehicle, that allow changing the position
and orientation coordinates of the quadcopter freely in the
three-dimensional space. The dynamics are described by

\[
\begin{align}
    \ddot{x} &= (\ddot{f} \sin \theta - \beta \dot{x}) \frac{1}{m} \\
    \ddot{y} &= (\ddot{f} \cos \theta \sin \phi - \beta \dot{y}) \frac{1}{m} \\
    \ddot{z} &= -g + (\ddot{f} \cos \theta \cos \phi - \beta \dot{z}) \frac{1}{m} \\
    \dot{\theta} &= \frac{\tau_x}{I_{xx}} \\
    \dot{\phi} &= \frac{\tau_y}{I_{yy}} \\
    \dot{\psi} &= \frac{I_{zz}}{l} (\dot{f} + f_2 - f_3 + f_4) 
\end{align}
\]

where the damping factor \(\beta\) takes into account realistic
friction effects, \(m\) denotes the mass of the vehicle, and \(I_{xx}, I_{yy}, I_{zz}\) are the moments of inertia around the body frame
axes \(x, y, z\), respectively. The nonlinear dynamical model
has twelve states (six positions and six velocities) and four
inputs (the motors voltages \(V_{M_i}\)), largely coupled through
the nonlinear relations in (2). The parameters used in this
paper are reported in Table 1.
2.2 Linear MPC for stabilization

We design a linear MPC controller to stabilize the quadcopter vehicle around a desired position in space and attitude. In particular, the desired position \((x_d, y_d, z_d)\) will be commanded by the upper-layer hybrid MPC controller described below in Section 2.3 for collision avoidance.

We first linearize the nonlinear dynamical model (2) around an equilibrium condition of hovering. The resulting linear continuous-time state-space system is converted to discrete-time with sampling time \(T_s\),

\[
\begin{align*}
\dot{x}_L(k+1) &= A x_L(k) + B u_L(k) \\
y_L(k) &= \xi_L(k)
\end{align*}
\]

where \(x_L(k) = [\theta, \phi, \psi, z, \dot{z}, \dot{\theta}, \dot{\phi}, \dot{\psi}, \ddot{z}]^T \in \mathbb{R}^{13}\) is the state vector, \(u_L(k) = [v_{n1}, v_{n2}, v_{n3}, v_{n4}]^T \in \mathbb{R}^4\) is the input vector, \(y_L(k) \in \mathbb{R}^{13}\) 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 additional state \(z_1 = \int (z - z_d)\) is included to provide integral action on the altitude \(z\), so that offset-free tracking of the desired setpoint \(z_d\) is guaranteed in steady-state. The integral action is mainly due to counteract effect of the gravity force acting against the force developed by the collective input \(f\).

The linear MPC formulation of the Model Predictive Control Toolbox for MATLAB [Bemporad et al., 2004b] based on quadratic programming is used to design the controller.

2.3 Hybrid MPC for collision avoidance

The proposed approach consists of constructing an abstract hybrid model of the controlled aerial vehicle and of the surrounding obstacles, and then use a hybrid MPC strategy for on-line generation of the desired position \((x_d, y_d, z_d)\). The closed-loop dynamics composed by the quadcopter and the linear MPC controller can be very roughly approximated by the discrete-time linear dynamical system

\[
\begin{align*}
x(k+1) &= a_1 x(k) + b_1 x_d(k) + \Delta x_d(k) \\
y(k+1) &= a_2 y(k) + b_2 y_d(k) + \Delta y_d(k) \\
z(k+1) &= a_3 z(k) + b_3 z_d(k) + \Delta z_d(k)
\end{align*}
\]

where \(\Delta x_d(k) = x_d(k) - x_d(k-1)\) is the input of the system representing the increment of desired \((x)\)-coordinate commanded at time \(k T_{hyb}\), \(T_{hyb} > T_s\) is the sampling time of the hybrid MPC controller, and \((x_d(k), y_d(k), z_d(k))\) is the output of the system, representing the actual position of the quadcopter. The parameters \(a_1, a_2, a_3, b_1, b_2, b_3\) in (4) are estimated by simulating the linear MPC controller designed in Section 2.2. The input increments \(\Delta x_d(k), \Delta y_d(k), \Delta z_d(k)\) are upper and lower bounded by a quantity \(\Delta\),

\[
\begin{align*}
-\Delta &\leq \begin{bmatrix} \Delta x_d(k) \\
\Delta y_d(k) \\
\Delta z_d(k)
\end{bmatrix} \leq \Delta
\end{align*}
\]

which is a tuning knob of the hybrid MPC controller, as it allows one to directly control the speed of maneuver of the quadcopter by imposing constraints on the reference derivatives.

Obstacles are modeled as polyhedral sets. To minimize the complexity of the overall hybrid prediction model described in this section, the \(i\)th obstacle, \(i = 1, \ldots, M\), is modeled as the tetrahedron

\[
A_{obs} k_i z(k) \leq B_{obs}
\]

where \(A_{obs} \leq B_{obs}\) is a fixed hyperplane representation of a reference tetrahedron, \(k_i\) is a fixed scaling factor, and \(z(k)\) is a reference point of the obstacle. In this paper we choose \(A_{obs}, B_{obs}\) such that the corresponding polyhedron is the convex hull of vectors

\[
\begin{bmatrix} 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 5/2 & 5/2 & 5/2 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
5/2 & 5/2 & 5/2 & 5/2
\end{bmatrix}
\]

which makes the reference point \((x_i, y_i, z_i, \theta, \phi, \psi)\) its vertex with smallest coordinates.

Equation (6) can be rewritten as

\[
A_{obs} k_i \begin{bmatrix} x(k) \\
y(k) \\
z(k) \end{bmatrix} \leq C_{obs}(k)
\]

where \(C_{obs}(k) = B_{obs} + A_{obs} k_i \begin{bmatrix} x(k) \\
y(k) \\
z(k) \end{bmatrix}\), \(C_{obs}(k) \in \mathbb{R}^4\), is a quantity that may vary in real-time. Although we model here the predicted evolution of \(C_{obs}\) as

\[
C_{obs}(k+h+1) = C_{obs}(k+h)
\]

non-constant dynamics might be used as well if estimates of obstacle velocities and/or accelerations are available.

Finally, to represent obstacle avoidance constraints, define the following binary variables \(\delta_{ij} \in \{0, 1\}, i = 1, \ldots, M, j = 1, \ldots, 4\)

\[
\delta_{ij}(k) = 1 \iff A_{obs} k_i \begin{bmatrix} x(k) \\
y(k) \\
z(k) \end{bmatrix} \leq C^i_{obs}(k)
\]

where \(j\) denotes the \(j\)th row (component) of a matrix (vector). The following logical constraints

\[
\forall i = 1, \ldots, M
\]

impose that at least one linear inequality in (7) is violated, therefore enforcing the quadcopter position \((x(k), y(k), z(k))\) to lie outside each obstacle.

Differently from Bemporad et al. [2009], we want to penalize here that vehicles fly too close to obstacles. To this end, each obstacle seen by a UAV is surrounded by a “safety area”, represented by a larger tetrahedron containing the one defined by (7). For such safety areas, we also introduce binary variables \(\delta_{ij}(k), \ell = 1, \ldots, S, \) as in (9), where \(S\) is the number of safety areas \((S \leq M\) in case of one UAV flying in an area with \(M\) obstacles), but without imposing the logical constraints as in (10), so permitting the UAV entering those areas. Such an event is penalized by defining for each “safety area” a variable \(\gamma_{ij}(k), \ell = 1, \ldots, S\).
\[ \gamma_{\ell}(k) = \begin{cases} 1 & \text{if } \bigwedge_{j=1}^{4} \delta_{\ell j}(k) = 1 \ \forall \ell = 1, \ldots, S \\ 0 & \text{otherwise} \end{cases} \quad (11) \]

that the controller will try to keep at zero. In this way, the vehicle will tend to avoid passing through the safety areas that have been set around the obstacles. Note that although \( \gamma_{\ell} \) can only assume values 0 and 1, we treat it as a real variable to ease the hybrid MPC computations that will be defined in the next paragraphs. The sampling time \( T_{\text{hyp}} \) must be chosen large enough to neglect fast transient dynamics, so that the lower and upper MPC designs can be conveniently decoupled. On the other hand, the obstacle avoidance constraint (10) is only imposed at multiples of \( T_{\text{hyp}} \). Note that an excessively large \( T_{\text{hyp}} \) may lead to trajectories that go through the obstacles during intersampling intervals \((kT_{\text{hyp}}, (k + 1)T_{\text{hyp}})\).

### 2.4 Hybrid model for UAV navigation in formation

The hierarchical structure described above for one quadcopter is extended to coordinate a formation of \( V \) cooperating UAVs, \( V > 1 \). We use a leader-follower approach with decentralized scheme to manage the formation: one of the vehicles is chosen as a leader to direct the formation following a prescribed path, and all the other vehicles, the followers, are commanded to maintain a constant relative distance reference from the leader. Each UAV is equipped with its own hybrid MPC controller and takes decisions autonomously, measuring its own state and the positions of the neighboring vehicles and obstacles, planning its own path under obstacle avoidance constraints. The whole 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). Moreover, each UAV must avoid collisions with the other vehicles in the formation, that are treated as obstacles, so that \( M \) accounts now for both real obstacles and other (neighboring) vehicles.

The overall hybrid dynamical model is obtained by collecting (4), (8), (9), (10), (11). These are modeled through the modeling language HYSDEL [Torrisi and Bemporad, 2004] and converted automatically by the Hybrid Toolbox for MATLAB [Bemporad, 2004] into mixed logical dynamical (MLD) form [Bemporad and Morari, 1999]

\[ \xi_{H}(k+1) = A_{H}(k) + B_{1}\Delta u_{H}(k) + B_{2}\delta(k) + B_{3}\gamma_{H}(k) \]  

\[ y_{H}(k) = C_{H}(k) + D_{1}\Delta u_{H}(k) + D_{2}\delta(k) + D_{3}\gamma_{H}(k) \]  

\[ E_{2}\delta(k) + E_{3}\gamma_{H}(k) \leq E_{4}\Delta u_{H}(k) + E_{5}\xi_{H}(k) + E_{6} \]  

\[ 12(c) \]

where \( \xi_{H}(k) = [x(k) \ y(k) \ z(k) \ C_{1}(k) \ \ldots \ C_{M}(k) \ x_{2}(k-1) \ y_{2}(k-1) \ z_{2}(k-1)]^{T} \in \mathbb{R}^{6+4M} \) is the state vector, \( \Delta u_{H}(k) = [\Delta x_{d}(k) \ \Delta y_{d}(k) \ \Delta z_{d}(k)]^{T} \in \mathbb{R}^{3} \) is the input vector, \( y_{H}(k) = [x(k) \ y(k) \ z(k)]^{T} \in \mathbb{R}^{3} \) is the output vector, \( \delta(k) \in \{0, 1\}^{4M+4S} \) and \( \gamma_{H}(k) \in \mathbb{R}^{3} \) are respectively the vectors of binary (defined in (9)) and continuous auxiliary variables. The inequalities (12c) include a big-M representation [Bemporad and Morari, 1999] of (9) and a polyhedral inequality representation of (10). Matrices \( A_{1}, C_{1}, E_{1}, E_{2}, E_{4}, E_{5} \) have suitable dimensions and are generated by the HYSDEL compiler. In order to design a hybrid MPC controller, consider the finite-time optimal control problem

\[ \min_{\{\Delta u(k)\}_{k=0}^{N_{H}-1}} \sum_{j=0}^{N_{H}-1} (y_{H}(k + j + 1) - y_{H}(k + 1) - y_{H}(k)) + \Delta u_{H}(k + j)R_{\Delta u_{H}}(k + j) + \gamma_{H}(k + j)Q_{\gamma}(k + j) \]  

s.t. MLD dynamics (12)  

constraints (5)  

where \( N_{H} \) is the prediction horizon, \( y_{H} = [\bar{x}_{i} \ \bar{y}_{i} \ \bar{z}_{i}]^{T} \) is the desired position (e.g., the position of the target or of the leader vehicle), \( \gamma_{H} \) are the auxiliary continuous variables with reference \( \gamma_{H} = [Q_{y} \ \theta_{y} \ \gamma_{y}] \).\( Q_{y} \geq 0, R > 0 \in \mathbb{R}^{3 \times 3} \) and \( Q_{\gamma} \geq 0 \in \mathbb{R}^{S \times S} \) are weight matrices.

The MLD hybrid dynamics (12) have the advantage of making the optimal control problem (13) solvable by mixed-integer quadratic programming (MIQP). At each time \( t = 0, T_{\text{hyp}}, \ldots, kT_{\text{hyp}}, \ldots \), given the current reference values \( y_{H}(t) \) and the current state \( \xi(t) \), Problem (13) is solved to get the first optimal input sample \( \Delta u^{*}_{H}(t) \), which is commanded as the increment of desired set-point \( [x_{d}, y_{d}, z_{d}] \) to the linear MPC controller at the lower hierarchical level.

For comparison purposes, we also consider an alternative centralized scheme, consisting of a single hybrid MPC controller based on an overall model of the dynamics and of the obstacles of the entire formation, and generating the references for all UAVs. Clearly, this centralized approach has the drawback of requiring the solution of a single MIQP optimization problem for the entire team, which is typically more intensive from a computation viewpoint than solving \( V \) smaller hybrid MPC problems.

### 3. POTENTIAL FIELDS METHOD

For comparing the hybrid MPC approach with other existing navigation schemes, we consider the 3D potential fields method proposed in Paul et al. [2008] for a formation of helicopters, adapted for the formation of quadcopters defined earlier. In this case, obstacles are treated as spherical, rather than tetrahedral. A potential field is generated for each UAV depending on the formation pattern, desired and actual position, and obstacle positions, for collision and obstacle avoidance and target tracking. The total field

\[ F_{\text{tot}} = F_{t} + F_{ca} + F_{oa} \]  

(14)

for each vehicle, used for generating the references for the lower-level stabilizing linear MPC, is composed by the three components \( F_{t} \) (target tracking), \( F_{ca} \) (collision avoidance), and \( F_{oa} \) (obstacle avoidance). The contribution for tracking the position of the target is \( F_{t} = K_{t} = 1 \) (for position), where \( K_{t} \) is the gain for target potential, \( t \) is the target position and \( p \) is the vehicle position. For the leading vehicle, the target is a given fixed position \( p_{t} \), for the followers is a given distance \( p_{t} \) from the leader. To avoid collisions, a safety space is defined around each vehicle or obstacle, defined as a sphere with positive radius \( r_{s\text{av}} \). The additional field component for vehicle \( i \), whose safety sphere is invaded by vehicle \( j \), is defined by

\[ F_{\text{tot}} = F_{t} + F_{ca} + F_{oa} \]  

(14)
where $K_{ca}$ is the gain for collision avoidance and $d_{ij}, i \neq j$, is the distance between vehicles $i$ and $j$. The total amount of the collision avoidance term is given by
\begin{equation}
F_{ca}(k) = \begin{cases} 
\frac{K_{ca}}{||d_{ij}||} - K_{ca} & \text{if } ||d_{ij}|| \leq r_{sav} \\
0 & \text{otherwise}
\end{cases} \tag{15}
\end{equation}

However, there is an increment of computational complexity: while with the decentralized scheme the average CPU performance is quite satisfactory: trajectories circumvent obstacles and settle at the target point, while maintaining the desired formation as much as the obstacles consent it.

The following parameters are employed for hybrid MPC: $\alpha_{1x} = \alpha_{1y} = 0.6$, $\beta_{1x} = \beta_{1y} = 0.4$ for the approximation of the lower level dynamics; $N_H = 10$ (prediction horizon), $T_{hyb} = 1.5$ s, and $\Delta = \frac{1}{4}T_{hyb}$; $k_1 = k_4 = \frac{1}{4}$, $k_2 = \frac{10}{37}$, $k_3 = \frac{\alpha}{10}$ are the scaling factors used to model the obstacles as tetrahedra, and for each of them we have the corresponding “safety area” with scaling factor $k_s = \frac{1}{4+k_1,15,20}$, $i = 1, ..., 4$; finally, matrices $Q_0 = 0.01 \cdot I_{3 \times 3}$, $Q_2 = 10 \cdot I_{4 \times 4}$ and $R = 0.1 \cdot I_{3 \times 3}$ are used as weights.

The initial positions of the UAVs are $p_L(0) \triangleq [x_1(0), y_1(0), z_1(0)], p_F1(0) \triangleq [x_{F1}(0), y_{F1}(0), z_{F1}(0)], p_F2(0) \triangleq [x_{F2}(0), y_{F2}(0), z_{F2}(0)]$ for the leader, $p_{F1}(0) \triangleq [x_{F1}(0), y_{F1}(0), z_{F1}(0)]$, and $p_{F2}(0) \triangleq [x_{F2}(0), y_{F2}(0), z_{F2}(0)]$. The results were obtained on a Core 2 Duo running MATLAB R2009b, the Model Predictive Control and the Hybrid Toolbox under MS Windows, using the MIQP solver of CPLEX 11.1 [ILOG, Inc., 2008].

**4. SIMULATION RESULTS**

The overall system is tested by cascading the linear MPC controller with the hybrid MPC designed for navigation, according to the hierarchical scheme of Figure 1. The simulation consists of avoiding four obstacles of different dimensions, placed along the path to the target point.

The linear MPC controller is tuned according to the following setup. Regarding input variables, we set $v_{Lj}^\text{min} = 0 V$, $v_{Lj}^\text{max} = 11.1 V$, $w_{Lj}^\text{max} = 0.1$, $\forall j = 1, ..., 4$, $i = 0, ..., N_L - 1$. For output variables we set a lower bound $z_{L1}^\text{min} = 0$ on altitude, and upper and lower bounds $y_{\theta L1-2}^\text{min} = \frac{\pi}{6}$ on pitch $\theta$ and roll $\phi$ angles. The output weights are $w_{\theta}^y = 0$, $w_{\phi}^y = 0$, $j \in \{1, 2\}$, on $\theta$ and $\phi$, $w_{\phi}^y = 1$, $j \in \{7, 8\}$, on $\theta$ and $\phi$, and $w_{\theta}^y = 10$ on the remaining output variables. The chosen set of weights ensures a good trade-off between fast system response and actuation energy. The prediction horizon is $N_L = 20$, the control horizon is $N_{L\text{u}} = 3$, which, together with the choice of weights, allow obtaining a good compromise between tracking performance, robustness, and limited computational complexity. The sampling time of the linear MPC controllers is $T_s = \frac{1}{30}$ s. The remaining parameters $\nu_{\text{min}}, \nu_{\text{max}}, \rho$ are defaulted by the Model Predictive Control Toolbox [Bemporad et al., 2004a].

**Fig. 3. Trajectories of formation with obstacle avoidance, hybrid approach**

The trajectories obtained by using the decentralized hierarchical hybrid + linear MPC are shown in Figure 3. The performance is quite satisfactory: trajectories circumvent obstacles without collisions, and finally the quadcopters settle at the target point, while maintaining the desired formation as much as the obstacles consent it.

**4.2 Centralized hybrid + decentralized linear MPC**

Next, we compare the results with the trajectories obtained by a single centralized hybrid MPC planner cascaded by the decentralized set of linear MPC controllers for stabilization. The performance is very similar, and trajectories are also very similar to those reported in Figure 3. However, there is an increment of computational complexity: while with the centralized scheme the average CPU
Table 2. Parameters for the potential fields method

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_t$</td>
<td>0.15</td>
</tr>
<tr>
<td>$r_q$</td>
<td>1.3</td>
</tr>
<tr>
<td>$K_{ca}$</td>
<td>10</td>
</tr>
<tr>
<td>$K_{av}$</td>
<td>40</td>
</tr>
<tr>
<td>$r_{obs}(1)$</td>
<td>2.5</td>
</tr>
<tr>
<td>$r_{obs}(2)$</td>
<td>3</td>
</tr>
<tr>
<td>$r_{obs}(3)$</td>
<td>3.5</td>
</tr>
<tr>
<td>$r_{obs}(4)$</td>
<td>2.5</td>
</tr>
</tbody>
</table>

time to compute the hybrid MPC action for set-point generation is about 0.073 s per time step ($T_{hyb} = 1.5$ s), with the centralized scheme is about 0.466 s.

4.3 Comparison with the potential fields method

Consider now the potential fields method described in Section 3 to generate on-line the desired positions, while maintaining the lower-level linear MPC controllers for stabilization and reference tracking. In this case the obstacles are modeled as spherical obstacles, rather than polyhedra. The parameters used for simulation are reported in Table 2.

The potential fields method has a lower computational complexity, with a CPU time to calculate the desired position in the order of milliseconds. Even if the performance of obstacle avoidance (see Figure 4) is satisfactory, it takes a longer time to reach the target. Moreover, it is necessary to impose an upper bound $z_F^{max} = 22.4$ on altitude in the linear MPC formulation, to avoid undesired overshoots due to fast variations of the references. Finally, to make a quantitative comparison of the different control strategies, we show different performance indices in Table 3 and compare them for the different navigation algorithms: decentralized hybrid MPC, centralized hybrid MPC, and potential fields method. We consider the following three indices defined on the simulation interval 25-300 s (i.e., 350-4200 samples):

$$J_{tt} = \sum_{k=350}^{4200} \| p_L(k) - p_t \|^2$$
$$J_{fpt} = \sum_{k=350}^{4200} \| p_L(k) - p_{F1}(k) - p_{a1} \|^2 + \| p_L(k) - p_{F2}(k) - p_{a2} \|^2$$
$$J_u = \sum_{k=350}^{4200} \| u(k) - u(k-1) \|_1$$

where $J_{tt}$ represents the target tracking Integral Square Error (ISE) index, $J_{fpt}$ the formation pattern tracking ISE index, and $J_u$ the absolute derivative of input signals (IADU) index for checking the smoothness of control signals [Raffo et al., 2010]. The indices are normalized with respect to the values obtained using centralized hybrid MPC. It is apparent that the hybrid MPC approach outperforms the potential fields method. Note also that the decentralized and the centralized hybrid MPC schemes have almost equal performance, actually the decentralized scheme is even slightly better. This may be due to the receding-horizon mechanism of MPC and to the fact

![Fig. 4. Trajectories of formation with obstacle avoidance, potential fields method](image)

Table 3. Comparison of different approaches

<table>
<thead>
<tr>
<th>Scheme</th>
<th>$J_{tt}$</th>
<th>$J_{fpt}$</th>
<th>$J_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>centralized hybrid MPC</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>decentralized hybrid MPC</td>
<td>-0.09%</td>
<td>-0.51%</td>
<td>-0.03%</td>
</tr>
<tr>
<td>potential fields</td>
<td>+210.32%</td>
<td>+294.30%</td>
<td>+212.75%</td>
</tr>
</tbody>
</table>

that the MPC weights were tuned for the decentralized approach and then used for both schemes.

In the simulations we assumed that the positions of the obstacles are known at each sample step. In more realistic applications with several obstacles it may be enough to only know the locations of the obstacles which are closest to the vehicle, for a threefold reason. First, because of the finite-horizon formulation, remote obstacles will not affect the optimal solution, and may be safely ignored to limit the complexity of the optimization model. Second, because of the receding horizon mechanism, the optimal plan is continuously updated, which allows one to change the maneuvers to avoid new obstacles. Third, in a practical context obstacles may be moving in space, and since such dynamics is not modeled here, taking into account remote obstacles in their current position has a weak significance. The number $M$ of obstacles to be taken into account in the hybrid model clearly depends on the density of the obstacles and the speed of the vehicle. Note that, depending on the sensor system on board of the vehicle, it may be even impossible to measure the position of remote obstacles.

5. CONCLUSIONS

In this paper we have shown how model predictive control strategies can be employed for autonomous navigation of formation of unmanned aerial vehicles, such as quadcopters. A linear MPC controller takes care of vehicles stabilization and reference tracking, and a hybrid MPC generates the path to follow in real-time to reach a given target while avoiding obstacles. The simulation results have shown the reduced computational load of the decentralized hybrid MPC scheme compared to the centralized one, and the better performance of hybrid MPC in comparison to other on-line planning methods like potential fields. Compared to off-line planning methods, such a feature of
on-line generation of the 3D path to follow is particularly appealing in realistic scenarios where the positions of the target and of the obstacles are not known in advance, but rather acquired (and possibly time-varying) during flight operations.

REFERENCES


