# MPC Strategies for Cooperative Guidance of Autonomous Vehicles

S. Bertrand, J. Marzat, H. Piet-Lahanier, A. Kahn, Y. Rochefort

**S. Bertrand, J. Marzat,**
**H. Piet-Lahanier, A. Kahn,**
**Y. Rochefort**
(Onera)

E-mail: sylvain.bertrand@onera.fr

DOI : 10.12762/2014.AL08-11

# MPC Strategies for Cooperative Guidance of Autonomous Vehicles

This paper presents a comprehensive framework for the cooperative guidance of a fleet of autonomous vehicles, relying on Model Predictive Control (MPC). Solutions are provided for many common problems in cooperative control, namely collision and obstacle avoidance, formation flying and area exploration. Cost functions of the MPC strategy are defined to ensure a safe collaboration between the vehicles for these missions. An efficient way to select the optimal cost with limited computation time is also provided. The performance of the proposed approach is illustrated by simulation results.

## Introduction

Missions such as large area surveillance or multiple target tracking may often prove tedious, potentially dangerous or cumbersome for a human operator. Using autonomous, or at least partly autonomous, vehicles could greatly contribute to making these missions feasible. However, their complexity may prove very demanding in terms of technological requirements for a single vehicle. Splitting the task into several subtasks makes it easier to fulfill the demands. It is thus necessary to determine how the subtasks are defined and to which vehicles they are attributed. Two approaches can be defended. The first one consists in defining subtasks that do not interfere, but whose collection leads to mission achievement. The second one is aimed at defining imbricated subtasks whose coordinated achievements would be at least equal, but possibly greater, than those of a complex but unique vehicle. This approach, known as cooperative tasking, requires coordination of the entire vehicle set to guarantee the satisfaction of the initial mission needs. The determination of specific control laws and estimation procedures are required to enable vehicles to perform cooperative tasks. This field of research has been very active since the 1980's and encompasses theories from various domains, such as game theory, artificial intelligence or distributed control. The numerous existing approaches vary according to the type of mission and the associated requirements in terms of constraints on formation flying [1], [2], communication exchange [3] or allocation of resources [4]. Many issues must be addressed, from the modeling of the cooperative set of vehicles, the definition and management of the allocated tasks and required information, and the definition of cooperative control strategies enabling the coordination and safety of these vehicles.

In this paper, attention is focused on the design of control laws, assuming that the necessary shared information is available. Various characteristics can be attached to cooperation control from a system-control perspective, e. g., they can be designed as implicit or explicit, based on regulated or reactive control law, obtained by centralized or decentralized control, with equality or hierarchy among the vehicles.

Implicit cooperation describes the behavior of individuals obeying a set of basic rules that results in a cooperative behavior. This constitutes one of the basic features of biomimetic flocking [5]. In explicit cooperation, mission allocation and guidance laws are defined for enhancing cooperative behavior (see e.g. [6]).

Reactive or regulated control laws translate into a long-term versus short-term design of the guidance law. A regulated control law is designed to guide the vehicles along trajectories that have been previously designed, e. g., using path searching procedures, such as A star or Dijkstra [7] or other algorithms [8]. Reactive control is achieved using the current and predicted states of the system, including the vehicles and the environments [9]. It is designed to provide a trade-off between the mission objective and its safety, for the current time or a limited time horizon. Centralized or distributed cooperative control refers to the location where the control takes place. It can be achieved within a unique control unit interacting with all vehicles [10] or computed by each vehicle [11]. The latter implementation presents the advantage of being more robust in case of failure of one of the vehicles of the fleet, at the cost of increasing the amount of vehicle embedded computation load. The selection of centralized or distributed control is also linked to the definition of hierarchy among the vehicles. They can be considered as equal in terms of decision for allocation or objective making, or some vehicles can obtain a superior status providing them with a higher decision weight. As in centralized computation, this hierarchy lightens the computational burden required for each vehicle of the fleet, at the cost of making the fleet more vulnerable to potential failure of the privileged vehicles.

This paper focuses on the design of reactive and distributed cooperative control laws. It can be addressed by designing a common criterion reflecting the mission objectives in terms of aim, safety assessment and so on. This criterion is evaluated according to each vehicle action and result, taking also into account the interaction between vehicles. Control laws can be thus derived by optimization of this criterion, relying on approaches such as model predictive control (MPC). This method has been used at Onera to define cooperative guidance laws for safely performing various cooperative missions with a fleet of autonomous Unmanned Arieal Vehicles (UAVs). The paper is organized as follows. In the next section, basic features of model predictive control are recalled and their extension to cooperative control is presented. Then the proposed MPC method is applied to the main types of cooperative missions for a fleet of autonomous vehicles (formation flight with obstacle avoidance, area exploration).

## Model predictive control approach for UAV cooperative guidance

MPC has been widely used for the guidance of UAVs in various contexts. UAV flocking and formation flight has been discussed in [9]. In distributed MPC [12]–[14], each vehicle computes its control inputs at each timestep as a solution of an optimization problem over the future predicted trajectory. For tractability reasons, finite prediction and control horizon lengths, respectively denoted as $H_p$ and $H_c$, are used.

The future control inputs and the resulting state trajectories of a vehicle $i$ are written as

$$\mathbf{U}_i = ((\mathbf{u}_i(k))^T, (\mathbf{u}_i(k+1))^T, \ldots, (\mathbf{u}_i(k+H_c-1))^T)^T$$
$$\mathbf{X}_i = ((\mathbf{x}_i(k+1))^T, (\mathbf{x}_i(k+2))^T, \ldots, (\mathbf{x}_i(k+H_p))^T)^T$$

When $H_c < H_p$, we assume that the control inputs are 0 after $H_c$ steps. Once the optimal input sequence $\mathbf{U}_i*$ has been computed, each vehicle communicates its predicted trajectory to the rest of the fleet and applies the first sample of the computed optimal control sequence $\mathbf{U}_i*(k)$. The optimization problems at time $k$ take the following form:

$$\begin{aligned} &\text{minimize} && J_i(\mathbf{U_i}, \mathbf{X_i}) \\ &\text{over} && \mathbf{U_i} \in \mathcal{U}_i^{H_c} \\ &\text{subject to} && \forall t \in [k+1; k+H_p], \mathbf{x}_i(t) \in \mathcal{X}_i \end{aligned} \quad (1)$$

$J_i$ is the cost function associated with vehicle $i$. The constraints coupling the dynamics of the vehicles, such as collision avoidance, are taken into account by means of a penalty factor in the cost function. At the next timestep, each vehicle searches for its solution of problem (1).

The cost function $J_i$ is composed of a sum of terms reflecting the objectives of the mission. These terms are detailed in the following sections.

### Vehicle model

The $N$ UAVs are assumed to be identical. For the sake of simplicity, the UAVs are assumed to be pointwise and their trajectories are considered to be two dimensional, in a horizontal plane. Note that extension to 3D motion is straightforward.

The state and control vectors for each vehicle $i$ are defined as:

$$\mathbf{x}_i = \begin{pmatrix} x_i \\ y_i \\ v_i \\ \chi_i \end{pmatrix} \text{ and } \mathbf{u}_i = \begin{pmatrix} u_i^{\omega} \\ u_i^{v} \end{pmatrix} \quad (2)$$

where $x,y$ is the vehicle position, $v$ is its speed amplitude and $\mathcal{X}$ is its direction. The model dynamics are:

$$\begin{cases} x_i(k+1) = x_i(k) + \Delta t.v_i(k)\cos\chi_i(k) \\ y_i(k+1) = y_i(k) + \Delta t.v_i(k)\sin\chi_i(k) \\ v_i(k+1) = v_i(k) + \Delta t.u_i^{v}(k) \\ \chi_i(k+1) = \chi_i(k) + \Delta t.u_i^{\omega}(k) \end{cases} \quad (3)$$

where $\Delta t$ is the sampling timestep and $(u_i^v, u_i^w)$ are the longitudinal and rotational accelerations. The constraints on the dynamics (3) and the control inputs are:

$$\begin{aligned} v_{min} \le v_i \le v_{max} \qquad & -\omega_{max} \le \omega_i \le \omega_{max} \\ -\Delta v_{max} \le u_i^{v} \le \Delta v_{max} \qquad & -\Delta\omega_{max} \le u_i^{\omega} \le \Delta\omega_{max} \end{aligned} \quad (4)$$

We summarize the dynamics and the constraints as $\mathbf{x}_i(k+1) = f(\mathbf{x}_i(k), \mathbf{u}_i(k))$ and $(\mathbf{x}_i, \mathbf{u}_i) \in \mathcal{X}_i \times \mathcal{U}_i$. Communication delays and ranges are not considered here, all of the UAVs are assumed to have access without delay to the exact state of every vehicle at all times.

### Costs used for all types of missions

The cost function $J_i$ is composed of a navigation cost $J_i^{nav}$, a safety cost $J_i^{safety}$ and a control cost $J_i^u$:

$$J_i(k) = J_i^{nav}(k) + J_i^{safety}(k) + J_i^u(k) \quad (5)$$

The formulation of each cost function is presented in the following subsections.

#### Navigation cost

The navigation cost $J_i^{nav}$ is aimed at regulating the speed of the vehicles and controlling the way in which they navigate to way-points. It is divided into five cost functions:

$$\begin{aligned} J_i^{nav}(k) = J_i^{nav,horiz}(k) + J_i^{nav,vert}(k) + J_i^{nav,direct}(k) \\ + J_i^{nav,final}(k) + J_i^{nav,fleet}(k) \end{aligned} \quad (6)$$

The first two cost functions, $J_i^{nav,horiz}$ and $J_i^{nav,vert}$, respectively defined by (7) and (8), are aimed at regulating the modulus of the horizontal component $\mathbf{v}^h$ of the velocity around a nominal value $v_n$ and the vertical component $v^z$ of the velocity around a zero value (making the UAV fly at constant altitude in a 3D case).

$$J_i^{nav,horiz}(k) = W^{nav,horiz} \cdot \sum_{n=k+1}^{k+H_c} \left( \left\| \widehat{\mathbf{v}_i^h}(n|k) \right\| - v_n \right)^2 \quad (7)$$

$$J_i^{nav,vert}(k) = W^{nav,vert} \cdot \sum_{n=k+1}^{k+H_c} \left\| \widehat{v_i^z}(n|k) \right\|^2 \quad (8)$$

Weighting coefficients $W^{\bullet}$ are tuned to set relative priorities between each aspect of the mission. A method to tune these coefficients has been proposed in [14].

The following two cost functions, $J_i^{nav,direct}$ and $J_i^{nav,final}$, are used to make the vehicle fly along a straight-line reference trajectory oriented toward the next way-point and to drive it closer to this way-point. The reference trajectory from the current position $\mathbf{p}_i(k)$ of vehicle $i$ at time $k$ to the next way-point $p$ is composed of reference points $\mathbf{p}_{i,p}^{ref}(n|k)\left(n \in [k+1, k+H_p]\right)$ located at positions that vehicle $i$ would reach at timestep $n$ if moving along a straight line to $p$ at nominal velocity $v_n$, regardless of any constraints. These reference points are defined by (9) and illustrated in figure 1. The resulting definition of cost $J_i^{nav,direct}$ is given by (10).

$$\mathbf{p}_{i,p}^{ref}(n|k) = \mathbf{p}_i(k) + (n-k).\Delta t.v_n.\frac{\mathbf{p}_i(k) - \mathbf{p}_p}{\left\|\mathbf{p}_i(k) - \mathbf{p}_p\right\|} \forall n \in [k+1, k+H_p] \quad (9)$$

$$J_i^{nav,direct}(k) = W^{nav,direct} \cdot \sum_{n=k+1}^{k+H_p} \left\| \widehat{\mathbf{p}_i}(n|k) - \mathbf{p}_{i,p}^{ref}(n|k) \right\|^2 \quad (10)$$

In order to steer the vehicle toward the next way-point by the end of the horizon of prediction, let us define a reference ball $B_{i,p}^{ref}(k)$, illustrated in figure 1, as the smallest ball around way-point $p$ that vehicle $i$ can hypothetically reach from its current position by moving directly toward this way-point at nominal velocity $v_n$. It is defined as $B_{i,p}^{ref}(k) = \left\{\mathbf{x} \middle\| \left\|\mathbf{x} - \mathbf{p}_p\right\| \le r\left(B_{i,p}^{ref}(k)\right)\right\}$, where the radius $r\left(B_{i,p}^{ref}(k)\right)$ is given by:

$$r\left(B_{i,p}^{ref}(k)\right) = \begin{cases} 0 & \text{if } d_{ip}(k) \le H_p.\Delta t.v_n \\ d_{ip}(k) - H_p.\Delta t.v_n & \text{otherwise} \end{cases} \quad (11)$$

where $d_{ip}(k) = \left\|\mathbf{p}_p - \mathbf{p}_i(k)\right\|$ denotes the current distance between vehicle $i$ and the way-point $p$. Using these definitions, the cost $J_i^{nav,final}$ is defined by

$$J_i^{nav,final}(k) = W^{nav,final} \cdot \left( \left\| \widehat{\mathbf{p}_j}(k+H_p \mid k) - \mathbf{p}_p \right\| - r\left(B_{i,p}^{ref}(k)\right) \right)^2 \quad (12)$$

Finally, the fifth cost function $J_i^{nav,fleet}$ is aimed at making the vehicles remain together as a fleet. Its definition penalizes the predicted distance $\hat{d}_{ij}(n|k) = \left\|\hat{\mathbf{p}}_j(n|k) - \hat{\mathbf{p}}_i(n|k)\right\|$ between vehicles $i$ and $j(i \ne j)$:

$$J_i^{nav,fleet}(k) = W^{nav,fleet} \times \sum_{\substack{j=1 \\ j \ne 1}}^{N} \sum_{n=k+1}^{k+H_p} \frac{1 + \tanh\left(\left(\widehat{d_{ij}}(n|k) - \beta_{ij}^f\right).\alpha_{ij}^f\right)}{2} \quad (13)$$

where coefficients $\alpha_{ij}^f$ and $\beta_{ij}^f$ are defined by

$$\alpha_{ij}^f = 6.\left(d_{loss}^v(ij) - d_{des}^v(ij)\right)^{-1} \quad (14)$$

$$\beta_{ij}^f = \frac{1}{2}.\left(d_{loss}^v(ij) + d_{des}^v(ij)\right) \quad (15)$$

The coefficient $d_{des}^v(ij)$ defines a desired distance between the vehicles within the fleet, whereas $d_{loss}^v(ij)$ is the maximum distance allowed between vehicles of the fleet. Vehicles $j(i \ne j)$ beyond this maximum distance are not considered by vehicle $i$ any more. This represents, for example, limited communication and/or sensing ranges.

The definitions of $\alpha_{ij}^f$ and $\beta_{ij}^f$ have been chosen in such a way as to obtain a nearly constant cost for distances lower than $d_{des}^v(ij)$ or greater than $d_{loss}^v(ij)$ (defined by a derivative lower than 0.05) and a symmetric behavior at borders.

The change in $J_i^{nav,fleet}$ with respect to the distance $d_{ij}$ is plotted in figure 2.

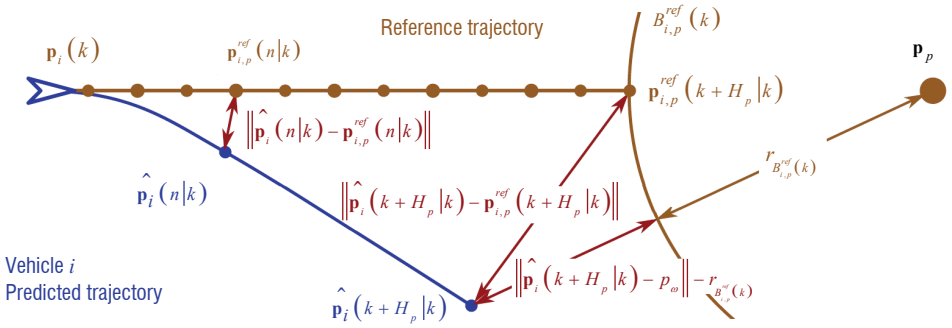



Fig. 1 - Representation of reference points $\mathbf{p}_{i,p}^{ref}$ and reference ball $B_{i,p}^{ref}$

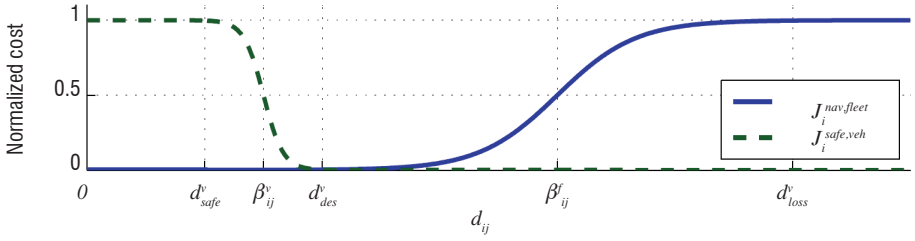



Fig. 2 - Flocking ( $J_i^{nav,fleet}$ ) and avoidance ( $J_i^{safe,veh}$ ) costs

### Safety cost

The safety cost $J_i^{safety}$ is aimed at avoiding collisions with obstacles and between vehicles within the fleet. It is composed of three cost functions:

$$J_i^{safety}(k) = J_i^{safe,veh}(k) + J_i^{safe,traj}(k) + J_i^{safe,obs}(k) \quad (16)$$

The first two costs deal with collision avoidance between vehicles, by respectively penalizing the predicted distance $d_{ij}$ between them and ensuring that the new predicted trajectory $\widehat{\mathbf{p}}_i\left(\begin{smallmatrix}k+H_p-1\\k+1\end{smallmatrix}|k\right)$ of vehicle $i$ remains close to the one transmitted to other vehicles at the previous iteration $\widehat{\mathbf{p}}_i\left(\begin{smallmatrix}k+H_p-1\\k+1\end{smallmatrix}|k-1\right)$:

$$J_i^{safe,veh}(k) = W^{safe,veh} \times \sum_{\substack{j=1\\ j\neq 1}}^{N} \sum_{n=k+1}^{k+H_p} \frac{1-\tanh\left(\left(\widehat{d_{ij}}(n|k) - \beta_{ij}^v\right).\alpha_{ij}^v\right)}{2} \quad (17)$$

$$J_i^{safe,traj}(k) = W^{safe,traj} . \sum_{n=k+1}^{k+H_p-1} \left\| \widehat{\mathbf{p}}_i(n|k) - \widehat{\mathbf{p}}_i(n|k-1) \right\|^2 \quad (18)$$

The shape parameters of the hyperbolic tangent function of equation (17), $\alpha_{ij}^v$ and $\beta_{ij}^v$, are defined by

$$\alpha_{ij}^v = 6.\left(d_{des}^v(ij) - d_{safe}^v(ij)\right)^{-1} \quad (19)$$

$$\beta_{ij}^v = \frac{1}{2}.\left(d_{des}^v(ij) + d_{safe}^v(ij)\right) \quad (20)$$

where $d_{safe}^v(ij)$ represents a desired safety distance between the vehicles. These shape parameters have been tuned according to the same considerations as previously explained for the definition of the safety cost. The change in $J_i^{safe,veh}$ with respect to the distance $d_{ij}$ is plotted in figure 2.

The third cost $J_i^{safe,obs}$ penalizes the predicted distance $\widehat{d_{io}}$ of vehicle $i$ to any obstacle $o$. It is defined as:

$$J_i^{safe,obs}(k) = W^{safe,obs} \times \sum_{o=1}^{N^o} \sum_{n=k+1}^{k+H_p} \frac{1-\tanh\left(\left(\widehat{d_{io}}(n|k) - \beta_{io}^o\right).\alpha_{io}^o\right)}{2} \quad (21)$$

where $N^o$ stands for the number of obstacles and the parameters $\alpha_{io}^o$ and $\beta_{io}^o$ are given by

$$\alpha_{io}^o = 6.\left(d_{des}^o(io) - d_{safe}^o(io)\right)^{-1} \quad (22)$$

$$\beta_{io}^o = \frac{1}{2}.\left(d_{des}^o(io) + d_{safe}^o(io)\right) \quad (23)$$

where $d_{des}^o(io)$ and $d_{safe}^o(io)$ are desired and safe distances to obstacles.

### Control cost

As traditionally defined in MPC, the control cost $J_i^u(k)$ is aimed at limiting the control effort and thus the energy consumption of vehicle $i$. It is defined by the following quadratic form:

$$J_i^u(k) = \sum_{n=k+1}^{k+H_c} \mathbf{u}_i(n|k)^T \begin{bmatrix} W^{u,v} & 0 \\ 0 & W^{u,\omega} \end{bmatrix} \mathbf{u}_i(n|k) \quad (24)$$

## Online computation of best cost

The MPC optimization problem (1) is a constrained nonlinear program, the solution of which cannot be found analytically. Numerical optimization must hence be used to approximate the solution.

Global optimization procedures based, for example, on interval analysis [15] or genetic algorithms [16] can be used, but may in practice be computationally prohibitive for real-time implementation. Numerical optimization methods, such as Sequential Quadratic Programming (SQP), Active Set or Interior Point methods, are thus generally preferred [17], [18]. Other methods suitable for MPC problems have also been developed [19]. Nevertheless, a global solution can be hard to find because of potential local minima. The computational time required for a MPC approach strongly depends on the parameterization of the control sequence. Low dimensional parameterizations have, for example, enabled successful applications to control systems with fast dynamics [20], [21]. Another solution consists in considering a finite set of predefined feasible control sequences, from which the one minimizing the cost function will be selected [22]. This last solution is used in this paper for implementation of the MPC strategy, based on [14].
This systematic search strategy has several main advantages over a traditional optimization procedure. Firstly, the computation load necessary to find a control sequence is constant in all situations leading to constant computation delay. The second advantage is that the systematic search strategy can be less sensitive to local minima problems, since the entire control space is explored. Finally, the systematic search requires no initialization of the optimization procedure.

The studied search procedure consists in defining, prior to the mission, a set $S$ of candidate control sequences that satisfy control constraints (4). At each timestep, the control problem (1) is solved using the proposed search procedure, as follows:

- using a model of the vehicle dynamics, predict the effect of each control sequence of the set of candidates $S$ on the state of the vehicle;
- remove from $S$ all of the candidate control sequences that lead to a violation of constraints on the state of the vehicle (4);
- compute the cost $J_i$ corresponding to each remaining candidate control sequence;
- select the control sequence that entails the smallest cost.

Since all of the candidates in the set $S$ will be evaluated, the computation load of associated predictions should be as limited as possible. A simple parameterization of the control sequence is therefore adopted, by considering a control input constant over the control horizon $H_c$ and then null over the remainder of the prediction horizon $H_p$. In addition, the distribution of the candidate control sequences is chosen so as to limit their number, while providing a good coverage of the control space.

The following three rules have been chosen:

- the set $S$ of candidates includes the extreme control inputs, to exploit the full potential of the vehicles;
- the set $S$ of candidates includes the null control input, to allow the same angular and linear velocities to be continued with;
- candidates are distributed over the entire control space, with an increased density around the null control input.

Constraints on the control inputs (4) can be translated into constraints on the norm of the horizontal component $\left(\left\|\mathbf{a}^h\right\| \le a^h_{\max}\right)$ and on the vertical component $\left(-a^z_{\max} \le a^z \le a^z_{\max}\right)$ of the acceleration of the vehicle. Therefore, it has been chosen to define the set $S$ in terms of accelerations as follows:

$$S = \left\{\left\{S^\omega \times S^v\right\} \cup (0,0)\right\} \times S^z \tag{25}$$

where $S^\omega$, $S^v$ and $S^z$ are respectively the sets of directions, modules and vertical components of the acceleration, defined by:

$$S^\omega = \left\{\frac{2.\pi.p}{\eta^\omega}\right\} \text{ with } p = 1 \text{ to } \eta^\omega \tag{26}$$

$$S^v = \left\{\frac{a^h_{\max}}{\left(\zeta^v\right)^p}\right\} \text{ with } p = 0 \text{ to } \eta^v \tag{27}$$

$$S^z = \left\{\pm\frac{a^z_{\max}}{\left(\zeta^z\right)^p}\right\} \cup \{0\} \quad \text{with } p = 0 \text{ to } \eta^z \tag{28}$$

$\zeta^v$ and $\zeta^z$ control the interval between two candidates; the number of candidates $N^c$, $N^\omega$, $N^v$, and $N^z$ in $S$, $S^\omega$, $S^v$, and $S^z$ respectively are deduced from $\eta^\omega$, $\eta^v$, and $\eta^z$ using (29) to (32).

$$N^c = \left(N^\omega.N^v + 1\right).N^z \tag{29}$$

$$N^\omega = \eta^\omega \tag{30}$$

$$N^v = \eta^v + 1 \tag{31}$$

$$N^z = 2.\eta^z + 1 \tag{32}$$

The resulting complete set $S$ is illustrated in figure 3. Using this set $S$, a vehicle can be aimed at any arbitrary way-point in some iterations. The minimal distance at which the vehicle can approach the way-point depends on the precision of the control, defined by the values of the $\zeta$• and $\eta$• parameters.

## Applications

Applications of the MPC strategy for three different types of missions are proposed in this section. The first one concerns the guidance of a fleet of quadrotor UAVs toward given objectives represented by way-points, while avoiding collisions with obstacles and between vehicles [14]. Exploration missions are then addressed by a cooperative grid allocation approach [23] and a cost-oriented approach [24]. The third type of mission is formation flying, for which an adaptable virtual structure is proposed along with the MPC approach [25].

### Guidance of a fleet toward predefined objectives

Many missions consist in making a group of several autonomous vehicles successively reach predefined objectives. These objectives may be defined in terms of a sequence of way-points, toward which the group of vehicles must be guided. Cooperation hence consists in sharing information (predicted trajectories) and controlling each vehicle of the group in such a way that the fleet as a whole can safely reach each of the waypoints.

The proposed cooperative MPC scheme is illustrated in this section for such a mission, where a group of $N=7$ vehicles must successively reach three way-points while avoiding collisions with obstacles and within the fleet. The vehicles considered are quadrotor UAVs, for which true dynamics include an inner loop for attitude control. Robustness to model mismatch of the MPC guidance strategy is therefore evaluated, since the prediction model used in the MPC guidance scheme consists in a 3D extension of (3). The quadrotor model and simulation results are presented in the next paragraphs.

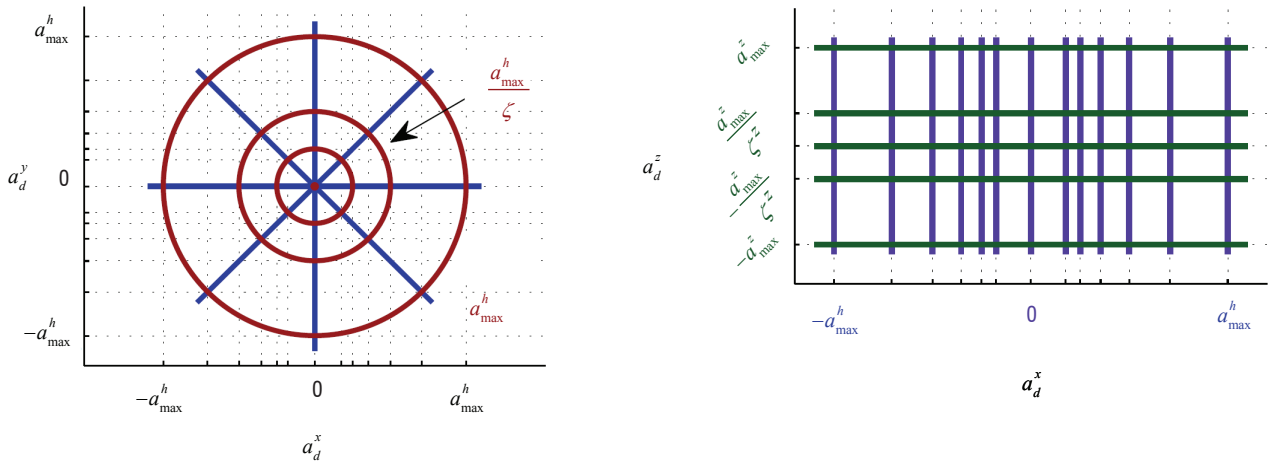



Fig. 3 - Illustration of the set $S$ of candidate control sequences, defined in terms of accelerations $\left(\zeta^v = 2,\ \zeta^z = 3,\ N^\omega = 8, N^v = 3 \text{ and } N^z = 5\right)$ - Projections on the $(x,y)$-plane (left picture) and on the $(x, z)$-plane (right picture)

### Quadrotor model and control strategy

Four-rotor helicopter models are derived from the description given in [26].

The rotors of each quadrotor $i$ are located at the four corners of a square, with opposite rotors rotating in the same direction and adjacent rotors rotating in opposite directions. The true control inputs of the rotors are the signals $u_i^1$ to $u_i^4$ , classically defined as:
$u_i^1$ : the resulting thrust of the four rotors (which controls the motion along the $z$-axis of the vehicle);
$u_i^2$ : the difference of thrust between the left and right rotors (which controls the roll $\varphi_i$ and hence contributes to the motion along the $y$-axis of the vehicle);
$u_i^3$ : the difference of thrust between the front and back motors (which controls the pitch $\theta_i$ and hence contributes to the motion along the $x$-axis of the vehicle);
$u_i^4$ : the difference of torque between the clockwise and anti-clockwise rotating rotors (which controls the yaw $\psi_i$ of the vehicle).

It is assumed that, at each iteration $k$, the value of the state composed of the position, attitude angles, and the linear and angular velocities, is available for the computation of vehicle $i$ control. The control strategy consists in applying the MPC guidance law based on a simplified 3D version of the prediction model (3), described in § "Vehicle model", to compute a desired acceleration vector $\mathbf{a}_{d,i}(k) = \left[a_{d,i}^x(k)\ a_{d,i}^y(k)\ a_{d,i}^z(k)\right]^T$.

This desired acceleration, along with a given desired value $\psi_{d,i}$ for the yaw, is then converted into vehicle control inputs $(k)$ to $u_i^4(k)$. Note that attitude control of the quadrotor is achieved by using the approach proposed in [26].

Illustrations of the response of the controlled vehicle to an acceleration step in the $x$ direction and to a yaw angle step are given in figure 4. The desired value is shown as a red dotted line, whereas the simulated response is shown as a plain blue line.

### Mission set-up and tuning parameters

In the simulated mission, the flock must successively reach three way-points, while avoiding obstacles and collisions. The vehicles must also travel as a group at nominal velocity $v_n = 2$ m.s$^{-1}$. Defining the $z$-axis downwards, the coordinate of the ground is $z = 0$ and the altitude of a vehicle is given by -$z$. At all times, the vehicles must fly between the altitudes of 0m (ground) and 25 m. These constraints are materialized with two obstacles of infinite dimensions, with a vertical safety distance of 2 m.

Initial positions of the vehicles are randomly chosen in the box defined by $x \in$[-205 m; -155 m], $y \in$[-45 m; 5 m], $z \in$[-15 m; -5 m]. Their initial velocities, attitudes and attitude derivatives are set to zero and the desired yaw angle $\psi_d$ is also set to zero for all vehicles throughout the entire mission. The guidance sampling time is $\Delta t = 0.5$ s.

Values of the constraints on velocity and acceleration are given in table I.

| | |
|---|---|
| $v_{\max}^h$ | 5 m.s$^{-1}$ |
| $v_{\max}^z$ | 1 m.s$^{-1}$ |
| $a_{\max}^h$ | 0.5 m.s$^{-2}$ |
| $a_{\max}^z$ | 0.25 m.s$^{-2}$ |

Table I - Velocity and acceleration constraints

Depending on the considered axis $(x, y, z)$, different values can be assigned to distances $d_{safe}^v(ij)$, $d_{des}^v(ij)$ and $d_{loss}^v(ij)$ introduced in § "Navigation cost" and "Safety cost" and respectively defining the safety and desired distances between vehicles and the distance threshold beyond which other vehicles are no longer considered. Corresponding ellipsoids $\varepsilon_{safe}^o(i)$, $\varepsilon_{des}^o(i)$ and $\varepsilon_{loss}^v(i)$ are therefore designed, as illustrated in figure 5. Their parameterizations are given in table II. The same applies for the distances $d_{des}^o(io)$ and $d_{safe}^o(io)$ to any obstacle $o$, to which similar ellipsoids are associated $\left(\varepsilon_{safe}^o(o)\ \varepsilon_{des}^o(o)\right)$ and whose parameters are also given in table II. The parameters of the search procedure used for the MPC strategy (lengths of horizons, size of the sets of control sequences and shape parameters) are given in table III and, finally, the tuning parameters of the objective functions are given in table IV. Let us recall that they define the relative importance of each component of the mission. Note that the weighting parameters of the control cost are defined here in terms of the horizontal and vertical components of the desired acceleration $\mathbf{a}_{d,i}$, which is considered as the control input computed by MPC for this case study.

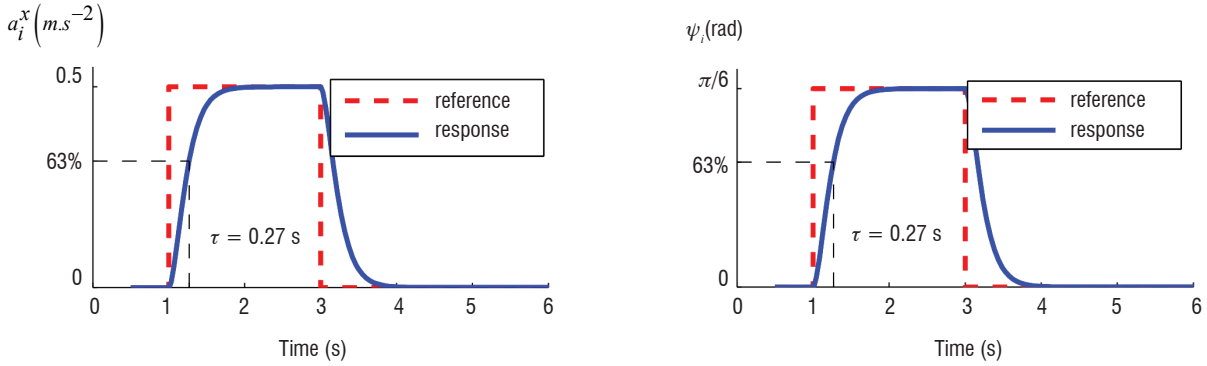



Fig. 4. Step response to a desired acceleration along $x$ of 0.5m.s$^{-2}$ (left picture) and to a desired yaw angle of $\pi/6$ rad (right picture)

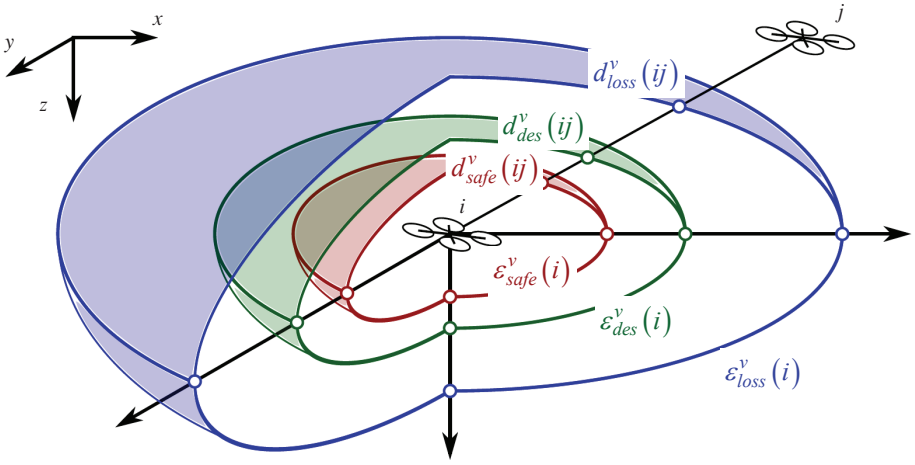



Fig. 5 - Safety $\left(\varepsilon^v_{safe}(i)\right)$, desired-locations $\left(\varepsilon^v_{des}(i)\right)$, and remoteness $\left(\varepsilon^v_{loss}(i)\right)$ ellipsoids around vehicle $i$

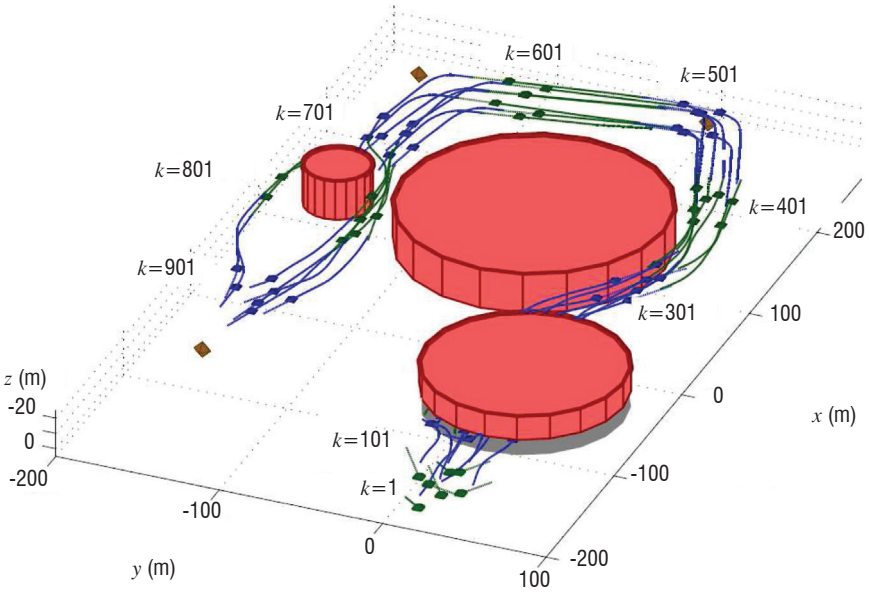



Fig. 6 - 3D view of the trajectories followed by the vehicles to complete their mission (way-points are represented by diamonds and obstacles by cylinders)

| ellipsoid | semi-axis length along $x$ | semi-axis length along $y$ | semi-axis length along $z$ |
|---|---|---|---|
| $safe$ | 10 m | 10 m | 5 m |
| $\varepsilon^v_{des}$ | 20 m | 20 m | 10 m |
| $\varepsilon^v_{loss}$ | 50 m | 50 m | 10 m |
| $\varepsilon^o_{safe}$ | 4 m | 4 m | 2 m |
| $\varepsilon^o_{des}$ | 8 m | 8 m | 4 m |

Table II - Parameters of the ellipsoids defining the characteristic distances between vehicles and to obstacles

| | | | |
|---|---|---|---|
| $H_c$ | 4 | $N^v$ | 3 |
| $H_p$ | 24 | $N^z$ | 5 |
| $N^\omega$ | 8 | $\zeta^v$ | 2 |
| $N^c$ | 125 | $\zeta^z$ | 3 |

Table III - Parameters of the search procedure

| | | | |
|---|---|---|---|
| $W^{u,h}$ | 2 | $W^{nav,horiz}$ | 10 |
| $W^{u,z}$ | 2 | $W^{nav,vert}$ | 2 |
| $W^{safe,veh}$ | 100 | $W^{nav,direct}$ | 10 |
| $W^{safe,obs}$ | 400 | $W^{nav,final}$ | 20 |
| $W^{safe,traj}$ | 0 | $W^{nav,fleet}$ | 50 |

Table IV - Weighting coefficients of the objective functions

| | |
|---|---|
| Mission success rate | 98.5% |
| Collision rate | 0% |
| % of loss of a vehicle | 1.5% |
| Mean computation time (std) | 18(1) ms |

Table V - Performance results of the MPC strategy over the 200 MC simulations

## Simulation results and analysis

The environment of the mission and the trajectories of the vehicles are presented in figure 6 for one realization of the mission (i.e., for given initial positions of the vehicles). This realization of the mission is considered to be a success, because all way-points have been successfully reached and vehicles have remained grouped together, while avoiding collisions. The first obstacle is avoided by flying under it, whereas the two other obstacles are avoided by turning around. Note that since $\psi_d = 0$, all vehicles are oriented along the $x$ axis (i.e., $\psi_i(k) = 0$ for all $i$ and $k$).

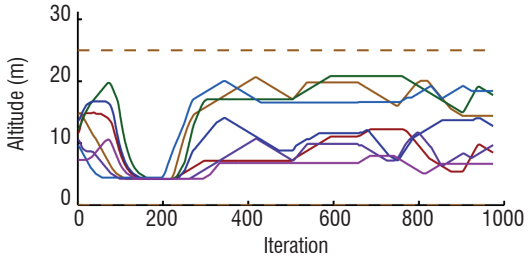



Fig. 7 - Change in the altitudes of the vehicles (the dotted lines represent the altitude constraints)

Altitude variations of the vehicles are presented in figure 7 and distances between the vehicles and to obstacles are presented in figure 8. As can be seen, the vehicles remained tightly grouped during the mission, except when they had to avoid obstacles, and spread over the vertical axis to form a tighter group while maintaining the desired distance between them. In addition, the vehicles always managed to avoid entering the safety ellipsoid of obstacles or other vehicles. Constraints on velocity and control inputs of the vehicles are satisfied throughout the mission, as can be seen in figures 9 and 10.

All of the aforementioned results concern one realization of the mission, i.e., one simulation corresponding to given initial positions of the vehicles. 200 Monte Carlo simulations have been run, randomly choosing these initial positions in the box defined by $x \in$ [-205 m; -155 m], $y \in$ [-45 m; 5 m], $z \in$ [-15 m; -5 m]. Table V provides the rate of success, collision and loss-of-a-vehicle (i.e., when the distance between two vehicles becomes greater than $d^v_{loss}$) over these 200 simulations. The mean value and standard deviation of the computation time (using Matlab on a standard PC) are also presented, over the 200 simulations. Note

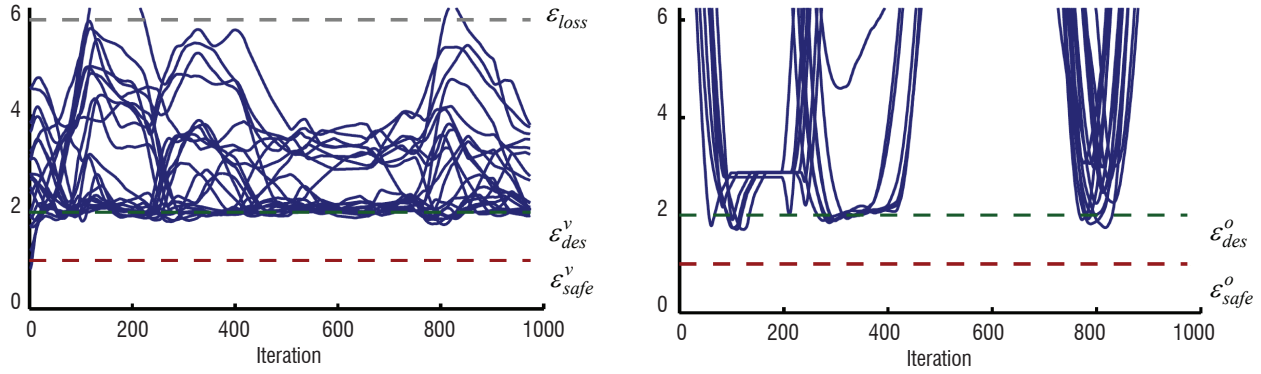



Fig. 8 - Distances between vehicles (left picture) and to each obstacle (right picture)

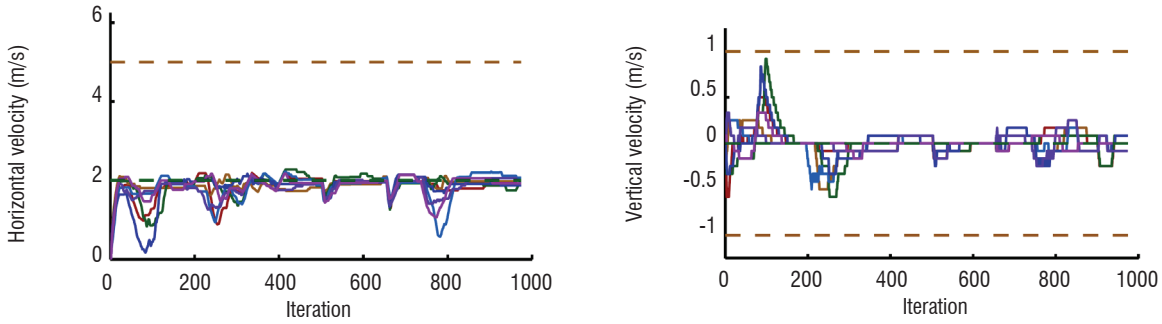



Fig. 9 - Constraints on the horizontal and vertical components of the velocity

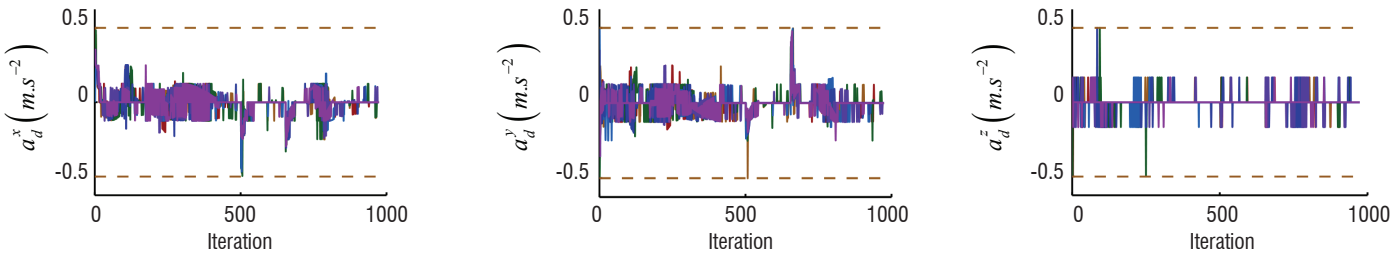



Fig. 10 - Components of the desired acceleration (control input)

that, for a given mission, the computation time remains constant throughout the whole mission.

Other studies on the influence of search parameters (number of control sequence candidates) and comparison to a traditional optimization procedure (SQP) have also been conducted [14], confirming the good performance and robustness of the proposed MPC strategy.

## Area exploration via cooperative grid allocation

The cooperative exploration problem addressed in this section consists in zone surveillance. The zone is described by a value grid, as in [27] or [28]. Each point of the grid must be visited at least once by one of the vehicles belonging to the fleet.

### Adaptation of navigation and objective criteria

The first step is to allocate the grid points among the vehicles. In order to maintain a distributed control design approach, the procedure suggested here consists in allocating to each vehicle two grid points selected at each iteration step among the remaining points, using the following selection criterion. This criterion, denoted $J^{attr}_{i,\{p_1,p_2\}}$, is built as the sum of the three following terms:

- $J^{pec}_{i,p_1}$ estimates the maneuvering cost of the vehicle to reach the first grid point of the allocated pair,
- $J^{pec}_{i,p_2|p_1}$ estimates the additional maneuvering cost for the vehicle to reach the second grid point, while progressing to the first allocated point,
- $J^{dev}_{i,\{p_1,p_2\}}$ estimates the additional cost for the vehicles that were initially directed to $p_1$ or $p_2$ and have to be redirected as they are allocated to vehicle $i$.

The first component of the criterion $J^{pec}_{i,p_1}$ is determined as a function of the distance between the first grid point and the predicted trajectory defined over the horizon of prediction. It is expressed as

$$J^{pec}_{i,p_1}(k) = \widehat{d^{\min}_{i,p_1}}(k) \tag{33}$$

$$\widehat{d^{\min}_{i,p_1}}(k) = \min_{n\in[k+1,k+H_p]} \left\| \widehat{\mathbf{p}}_i(n) - \mathbf{p}_{p_1} \right\| \tag{34}$$

The criterion $J^{pec}_{i,p_2|p_1}$ expression varies, whether the previous grid point $p_1$ is reached before, after half of the duration of the prediction horizon, or not reached at all. In the first case, $J^{pec}_{i,p_2|p_1}$ is defined as

$$J^{pec}_{i,p_2|p_1}(k) = \widehat{d^{\min}_{i,p_2}}(k) \tag{35}$$

In the second case, the former expression is transformed into

$$J^{pec}_{i,p_2|p_1}(k) = \left(1+\lambda_\theta.\widehat{\theta^{rel}_{i,p_2}}\left(k+H_p\right)\right).\widehat{d_{i,p_2}}\left(k+H_p\right) \tag{36}$$

where $\widehat{d_{i,p_2}}\left(k+H_p\right)$ is the distance between the predicted position of the vehicle at $H_p$ and the second grid point, and $\widehat{\theta^{rel}_{i,p_2}}\left(k+H_p\right)$ is the variation of the predicted attitude of the vehicle $\widehat{\chi_i}\left(k+H_p\right)$ and the direction of the line of sight $\arg\left(\mathbf{p}_{p_2} - \widehat{\mathbf{p}}_i\left(k+H_p\right)\right)$.

In the latter case, the criterion is of similar form:

$$J^{pec}_{i,p_2|p_1}(k) = \left(1+\lambda_\theta.\widehat{\theta^{rel}_{i,p_2|p_1}}\left(k+\frac{H_p}{2}\right)\right).d_{p_1,p_2} \tag{37}$$

where $d_{p_1,p_2}$ is the distance between $p_1$ and $p_2$.

The last component $J^{dev}_{i,\{p_1,p_2\}}$ of the cost $J^{attr}_{i,\{p_1,p_2\}}$ requires each vehicle to establish a list of the potential pair of allocated points, to cross-check with the other vehicles whether they are at risk of being deviated from their initial choice. The list contains the available pairs of grid points classified by decreasing order of sum of $J^{pec}_{i,p_1}$ and $J^{pec}_{i,p_2|p_1}$.

Simultaneously to the selection of objectives, a navigation criterion must be computed. The functional developed for this purpose is derived from the cooperative MPC strategy presented in § " Guidance of a fleet toward predefined objectives". The differences between way-point guidance and way-point allocation are, first, that the way-points describing a grid zone are closer to each other than the way-points used to indicate a global path to the fleet and, second, that the trajectories are defined to be close to the grid points instead of being directed towards them. Hence, the navigation criterion $J^{nav,final}$ must be adapted accordingly. In this context, it depends on the minimal distance $\widehat{d^{\min}_{i,p_1}}(k)$ defined by (34) between predicted positions of the vehicle and the way-point that it must explore during the prediction horizon. The resulting expression of the criterion is

$$J^{nav,final}_i(k) = \begin{cases} W^{nav,final} \times \left(\widehat{d^{\min}_{i,p_1}}(k)\right)^2 \\ if\ d_{i,p_1}(k) \le H_p.\Delta t.v_n \\ W^{nav,final} \times \left(d\left(\widehat{\mathbf{p}}_i\left(k+H_p\right), B^{ref}_{i,p_1}(k)\right)\right)^2 \\ otherwise \end{cases} \tag{38}$$

### Application example

Figure 11 presents the trajectories obtained for an exploration mission on a grid zone, realized by four vehicles. The duration of the mission is 400 s. At the end of the mission, all vehicles must reach an exit point of coordinates $x$ =-200, $y$ =250.

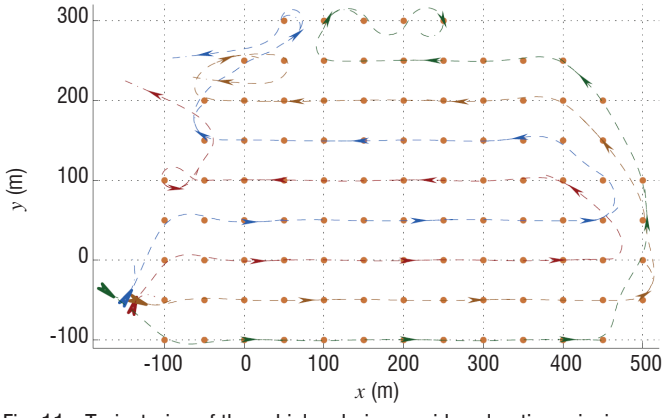



Fig. 11 - Trajectories of the vehicles during a grid exploration mission

It illustrates some of the effects of the approach described. At the beginning, the vehicles separate to reach a different row of grid points. While the number of remaining grid points remains high enough to limit multiple allocations, the vehicles follow straight lines. When allocation is sparse, the trajectories may present oscillations, since the evaluation and comparison of costs require increased communications and the updating of allocated grid points.

## Area exploration via a cost-oriented approach

Zone watching is defined in this section as a cooperative problem, where a number of autonomous vehicles must explore a wide area in a limited amount of time without any way-point defined in advance. In addition to zone coverage, the dynamical allocation of exit locations is considered.

Each vehicle defines its own trajectory online to achieve the cooperative mission objectives:

- maximize the cumulated area covered,
- allocate and reach exit points at the end of the mission,

while respecting the constraints:

- collision avoidance,
- limited mission time,
- limited number of vehicles at a given exit.

The choice of optimal control entries should thus take into account four main aspects: collision avoidance, minimum control energy, map exploration and exit point assignment. The associated global cost function for this application is

$$J_i = J_i^{safety} + J_i^{u} + J_i^{expl} + J_i^{nav,direct} \tag{39}$$

The costs $J_i^{safety}$ to avoid collisions and $J_i^{u}$ to limit the energy spent by the vehicles are those defined respectively in § "Safety cost" and "Control cost". The cost $J_i^{expl}$ is specific to the exploration problem considered (see next §). The cost $J_i^{nav,direct}$ (defined in § "Navigation cost") is used to guide the vehicle to its allocated exit point, which is computed online, as indicated in § "Exit point (re)allocation". A dynamic weight gives more importance to this last cost when the mission time approaches its limit (§ "Weighting of the functions").

### Zone coverage

The cost function $J_i^{expl}$ should reflect the gain in terms of map exploration for a potential trajectory. Each vehicle is assumed to have a seeker capability, described by a function $f_{expl}$ of the relative position between the observed point and the vehicle.

The cooperatively explored area at time $k$ is:

$$\Omega = \bigcup_{\substack{t=1,\dots,k \\ i=1,\dots,N}} \mathcal{D}_i(t) \tag{40}$$

where $\mathcal{D}_i(t)$ is the sensing footprint of vehicle $i$ at timestep $t$. Since this representation is impractical, the mission field is approximated as a grid with spacing $d_{grid}$. A matrix $G$ stores the level of exploration of each square of the grid. Each element $G_{l,m}$ (where $l$, $m$ are the integer coordinates of the square in the grid) ranges between 0 when no vehicle has explored this location and 1 when it has been entirely observed. Each vehicle stores a copy of this exploration map and updates it with the information from the rest of the fleet. The precision of the representation only depends on the parameter $d_{grid}$. When a vehicle comes at a distance $d$ from the center of square $(l, m)$, the exploration level is updated as

$$G_{l,m}^{+} = \max\left(G_{l,m}, f_{expl}(d)\right)$$

The exploration index is increased only if the vehicle is close enough. The function $f_{expl}$ is chosen to be continuous and identically 0 for $d > r_{sensor}$. Here,

$$f_{expl}(d) = \begin{cases} 0 & if\ d \geq r_{sensor} \\ \frac{1}{2}\left(1+\cos\frac{\pi d}{r_{sensor}}\right) & if\ d < r_{sensor} \end{cases}$$

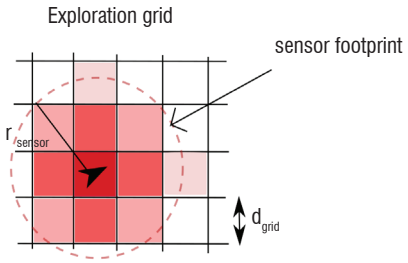



Fig. 12 - Illustration of exploration cost: colors reflect the exploration level

The cost function rewards trajectories that cooperatively increase the global level of exploration of the map. It is defined as

$$\begin{aligned} J_i^{expl} &= -W^{expl} \sum_{l,m} \left(G_{l,m}(k+H_p) - G_{l,m}(k)\right) \\ &= -W^{expl}\, \mathbf{1}^T.\left(G(k+H_p) - G(k)\right).\mathbf{1} \end{aligned} \tag{41}$$

where $G(k + H_p)$ is the predicted exploration map associated to the vehicle trajectory and $\mathbf{1}$ is the vector whose components are all 1. This cost function represents the total increase of the global exploration level resulting from a predicted trajectory. Since the vehicles share information, flying in already explored zones is therefore penalized.

### Exit point (re)allocation

Two cases are studied. In the first one, the number of vehicles $N$ is identical to the number of exits $n_t$ and a given exit point $c_j$ can shelter at the most one vehicle. In the second, the number of vehicles exceeds the number of exits and at most $n_{max}$ vehicles can reach a given exit point.

*Case $n_t = N$*

The aim is to define a cost that balances the distance to the exit and the cost in the control inputs (in other words, penalizes deviations from the trajectory to an exit point $c_j$ of position $\mathbf{p}_{cj}$). This cost will serve as a measure of the interest for a given vehicle to go to an exit and will support the decision of dispatching the vehicles to the exits. Therefore, it must discriminate efficiently between different vehicles aiming for the same exit.

For each pair {vehicle, exit point}, the cost function

$$J_{i,j}^{af}\left(\mathbf{u}_i(k), \mathbf{x}_i(k)\right) = \sum_{t=k}^{k+H_p-1} W^{d} \left\| \mathbf{p}_i(t) - \mathbf{p}_{cj} \right\|^2 \tag{42}$$

is minimized and an assignment cost matrix $R = \left(r_{ij}\right)_{\substack{i=1,\dots,N \\ j=1,\dots,n_t}}$ is obtained.

$$r_{ij} = \min_{\mathbf{u}_i} J_{ij}^{af}\left(\mathbf{u}_i, \mathbf{x}_i\right)$$

The optimal assignment is obtained by the Hungarian algorithm [29].

*Case $n_t < N$*

In this case, at the most $n_{max}$ vehicles can go to the same exit point at the end of the mission. The $(N \times n_t)$ matrix $R$ is built. A basic consensus mechanism is used to find a good admissible assignment: each vehicle forms a list of wishes based on its cost evaluations. These costs are centralized and whether no more than $n_{max}$ vehicles aim for each exit is checked. In case of conflict, the admissible alternative exits are considered. The minimizing costs among these are chosen for each conflicting vehicle, consecutively.

In these two cases, the construction of the cost matrix is decentralized but information must be centralized to perform the actual assignment.

The vehicles are now able to explore a zone and reach an exit at the end of the allocated time. The final constraints on the positions require a terminal allocation at all times, merely to ensure satisfaction of the constraints on the maximal number of vehicles for each exit. Nevertheless, if enough time remains, the vehicles should focus on exploration. Therefore, the initial assignment could be reconsidered after some time: a reallocation of the vehicle may prove beneficial. One option is to repeat the assignment procedure presented in the previous subsection during the mission. However, it could lead to an undesirable situation where, in order to decrease the total cost, the global optimization assigns to a vehicle an exit that cannot be reached before the end of the mission. Consequently, a penalty linking the time needed to reach the exit and the remaining mission time is added. It is expressed as follows:

$$s_{ij} = \begin{cases} 0 & \text{if } T \geq T_{safe} \\ \left(\dfrac{T_{safe} - T}{T_{safe} - T_{danger}}\right)^2 & \text{if } T < T_{safe} \end{cases} \quad (43)$$

where $T$ is the remaining time, $T_{danger} = \dfrac{\left\|\mathbf{p}_i - \mathbf{p}_{cj}\right\|}{v_n}$ and $T_{safe} = f_{safe} T_{danger} + T_{margin}$, $\mathbf{p}_{cj}$ is the position of the exit, and $T_{margin}$ and $f_{safe}$ are predefined parameters. The matrix used for the global assignment is $R' = \left(r'_{ij}\right)_{\substack{i=1,\dots N \\ j=1,\dots,n_t}}$ with $r'_{ij} = r_{ij} + s_{ij}$. The continuous variation of the penalty prevents vehicles from choosing unreachable exit points, provided that the reallocation is performed often enough. Repeating the allocation procedure frequently represents a large computational load, therefore instead of using the nonlinear dynamical model a simple linear model (double integrator) is used to approximate the dynamics. This does not significantly deteriorate the performance, because only estimates of the costs to go are required in order to choose a reasonable assignment. The linear approximation and the constraint translation is based on [13] and the reallocation can only be repeated at large time intervals.

**Weighting of the functions**

Each penalty function and its subcomponents are weighted with a coefficient $W\bullet = k\bullet . w\bullet$, with $k\bullet$ a normalization coefficient and $w\bullet$ a weighting coefficient. The $k\bullet$ (Table VI) coefficients are chosen so that without weighting, the worst case cost would be around 1.

| | | | |
|---|---|---|---|
| $k^{safety}$ | $\dfrac{2}{H_p}$ | $k^{expl}$ | $\left(\dfrac{2 r_{sensor} . v_n . H_p}{d_{grid}}\right)^{-1}$ |
| $k^{u}$ | $\dfrac{1}{\left(H_c . v_{\max} . \omega_{\max}\right)^2}$ | $k^{nav}$ | $\dfrac{1}{\overline{dist}^2}$ |

Table VI - Renormalization coefficients

Note that $k^{nav}$ is chosen so that exit allocation can be fairly performed between the vehicles: $\overline{dist} = \dfrac{1}{n_t . N} \sum_{i=1}^{N} \sum_{j=1}^{n_t} \left\|\mathbf{p}_i - \mathbf{p}_{c_j}\right\|$ is the average distance between vehicles and exits.

Since the total time allocated for the mission is known, it is preferable to rejoin the exit point only when the vehicles run out of time. A dynamic weighting procedure is proposed: the exploration and exit rejoining costs are weighted with respect to the difference between the estimated time to reach the exit and the actual remaining time. A scheme based on [30] is adopted: the exploration of the map is initially favored in the cost function, whereas exit points progressively take more importance in the cost function as the remaining time decreases. This translates into the algorithms by means of balancing coefficients $C^{expl}$, $C^{u}$ which multiply $W^{expl}$, $W^{u}$ as reported in Algorithm 1.

Algorithm 1: Calculation of the weighting coefficients

1) Compute $d = dist(\mathbf{p}_i, \mathbf{p}_{ci})$ distance between vehicle $i$ and its exit point.

2) Compute $T_{danger} = \dfrac{d}{v_n}$, the estimated minimal time to reach the exit assuming a straight path and nominal speed. Compute $T_{safe} = f_{safe} . T_{danger} + T_{margin}$, an overestimate of $T_{safe}$ considered as comfortable to reach the exit.

3) Compute:

$$C^{expl} = \begin{cases} 0 & \text{if } T \leq T_{danger} \\ \dfrac{T - T_{danger}}{T_{safe} - T_{danger}} & \text{if } T_{danger} < T \leq T_{safe} \\ 1 & \text{if } T > T_{safe} \end{cases}$$

$$C^{u} = \begin{cases} 1 & \text{if } T \leq T_{danger} \\ \dfrac{C_{\min}\left(T - T_{danger}\right) + T_{safe} - T}{T_{safe} - T_{danger}} & \text{if } T_{danger} < T \leq T_{safe} \\ C_{\min} & \text{if } T > T_{safe} \end{cases}$$

## Simulation results

Simulation parameters are grouped in table VII. The various requirements of the mission are first illustrated individually and quantitative simulation results are then given.

*Collision avoidance and exit rejoining*

The vehicles are positioned so that they have to cross paths to reach their exit points. No reallocation is allowed and map exploration is not taken into account. The dashed lines denote the past trajectories of the vehicles, whereas the dotted short lines depict the predicted trajectories at the current time. The circles denote the danger zones around the vehicles. Figure 13 shows that vehicles can reach agreements, even in complex situations, to cross ways without endangering themselves or the other vehicles.

| | | | | | |
|---|---|---|---|---|---|
| $v_{min}$ | 0.3 | $v_n$ | 0.7 | $v_{max}$ | 1.0 |
| $\Delta\omega_{max}$ | 0.1 | $\Delta v_{max}$ | 0.1 | $\omega_{max}$ | 0.5 |
| $d^v_{safe}$ | 4 | $d^v_{des}$ | 8 | $w^{safety}$ | 5 |
| $w^{tt}$ | 0.5 | $w^{nav,horiz}$ | 1 | $w^{expl}$ | 2 |
| $w^{el}$ | 2 | $f_{safe}$ | 1.1 | $C_{min}$ | 0.2 |
| $H_c$ | 3 | $H_p$ | 21 | $r_{max}$ | 5 |
| $T_{margin}$ | 15 | $f_{safe}$ | 1.1 | $d_{grid}$ | 2.5 |

Table VII - Simulation parameters for area exploration by cost-oriented approach

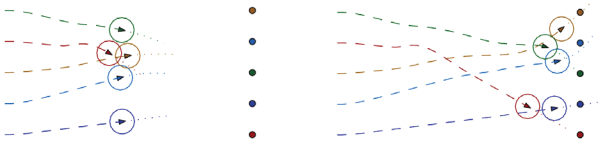
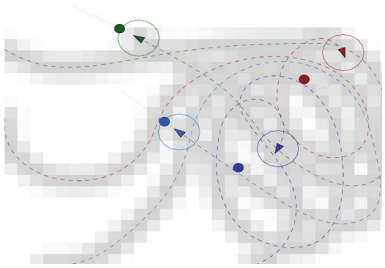
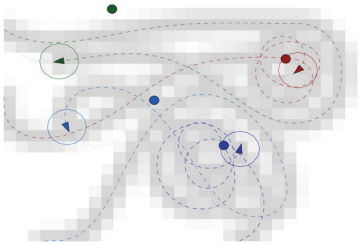


Fig. 13 - Illustration of the collision avoidance

*Map exploration and exit assignment*



(a) Weighting factors are dynamic: exploration is favored first and the exiting cost progressively prevails



(b) Weighting factors are constant

Fig. 14 - Comparison of different exploration strategies: the colors of the vehicles correspond to the assigned target

Map exploration and exit assignment are illustrated with a 4-vehicle scenario presented in figure 14. Exits are chosen randomly for each vehicle and no reallocation is allowed. It compares the behavior of the vehicles in two different settings: (a) dynamic weighting of exploration and exit assignment with respect to remaining time versus (b) constant weights. The main difference is that, in case (a), the vehicles can go far away from their exit point as long as time remains and, consequently, it is easier for them to find new zones to explore, while in case (b) vehicles tend to stay close to their exit point.

*Dynamic reallocation*

Dynamic reassignment is illustrated in figure 15. The current assignment in the figures is depicted by matching colors. In this particular instance, the vehicle beginning in the top left corner keeps his initial assignment during the mission, whereas the two others do not. One of them first changes its exit, whereas the last one also changes its plan later on.

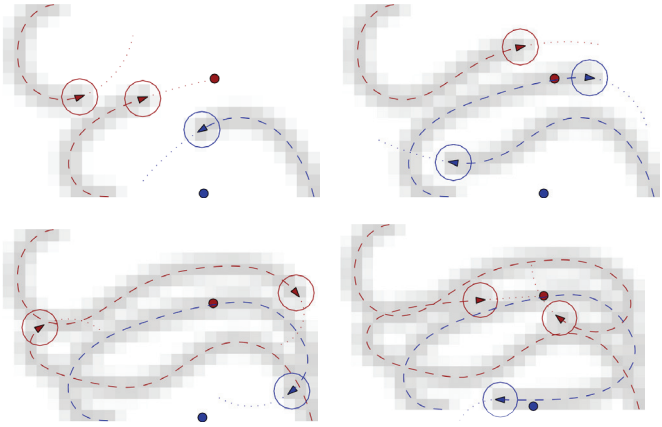


Fig. 15 - Online reassignment of the exit points: the colors of the vehicles correspond to the assigned target

*Performances*

To evaluate the performance of the strategies, a set of 70 missions was simulated with different configurations. The settings were:

- No exploration is considered: each vehicle chooses an exit and rejoins it as soon as possible (A)
- Exploration is considered, but weightings of exploration and exit rejoining in the cost function are fixed throughout the mission (B)
- Exploration is valued at the beginning of the mission and exit reaching progressively becomes the dominant cost (C)
- Configuration is identical to (C) but reassignment is granted (D)

In each mission, a 78 m x 78 m field is explored by 4 vehicles, with a mission time of 300s. The position of the vehicles and the 4 exit locations are chosen randomly for each run. The mission is performed for the 4 configurations and the results are given in table VIII. Expl(%) is the portion of the map that has been explored during the mission. It takes into account both the number of squares explored and their respective level of exploration. Danger(%) gives the ratio between the time during which a dangerous situation has occurred and the mission duration, that is, when 2 vehicles come closer than a distance of $d^v_{safe}$ at some point during the mission. $d_{exit}$ gives the average distance of the vehicles to their exit targets at the end of the mission. We can observe that map exploration costs allow a better exploration and dynamic weighting increases the efficiency of the exploration even more, as expected. However, it also

increases the number of dangerous situations: adding the exploration cost increases the chances of several vehicles coming into the same zone and therefore increases the collision risks. Furthermore, dynamical reallocation does reduce these risks significantly, while preserving the exploration efficiency. The results presented show that a global complex mission can be satisfactorily fulfilled by using a short-sighted and distributed control architecture. The proposed method renders the problem tractable and allows the actions to be taken online to be chosen.

| | Expl (%) | danger (%) | $d_{exit}\,(m)$ |
|---|---|---|---|
| A | 20.9 (3.0) | 1.4 | 4.2 (0.2) |
| B | 44.4 (5.0) | 5.7 | 8.4 (2.7) |
| C | 58.3 (3.2) | 11.4 | 6.2 (1.7) |
| D | 59.1 (3.3) | 8.5 | 6.5 (2.5) |

Table VIII - Simulation results (standard deviations are given in brackets)

## Formation flying using an adaptable virtual structure

Another guidance law is derived in this section to achieve formation flight toward a way-point for a fleet of autonomous vehicles. The formation is now defined by a virtual geometrical structure - here, an ellipse - that can modify its shape and orientation to avoid collision with obstacles in the environment. The proposed guidance law is divided into two layers, with a MPC scheme at each level. The higher layer controls the structure itself, to fulfill the goals and constraints of the required mission. The trajectory of the fleet is built on-line using this layer, as well as the adaptation of the structure to the environment. The lower layer controls the vehicles, so as to attract and keep them within the structure without side collision.

### Virtual structure control

The first layer of the guidance law generates the change in the virtual structure and adapts its shape so that it does not collide with the obstacles on its way to a way-point. It has been chosen to describe the formation shape as an ellipse, represented only by its center and characteristic matrix (this description can be steadily extended to that of an ellipsoid in 3D).

*Model of the virtual structure*

An ellipse with center $\mathbf{p}_c = [x_c\ y_c]^T$ and characteristic matrix $\mathbf{M}$ is defined by all points $\mathbf{p} = [x\ y]^T$ such that

$$(\mathbf{p}-\mathbf{p}_c)^T\,\mathbf{M}^{-1}(\mathbf{p}-\mathbf{p}_c) \le 1 \tag{44}$$

The characteristic matrix M of the ellipse can be written as

$$\mathbf{M} = \begin{bmatrix} \cos\chi_e & -\sin\chi_e \\ \sin\chi_e & \cos\chi_e \end{bmatrix} \begin{bmatrix} a^2 & 0 \\ 0 & b^2 \end{bmatrix} \begin{bmatrix} \cos\chi_e & -\sin\chi_e \\ \sin\chi_e & \cos\chi_e \end{bmatrix}^T \tag{45}$$

where the parameters of the ellipse are:

- $\chi_e$, the angle between the first principal axis and the horizontal;
- $a$, the length of the first principal axis of the ellipse;
- $b$, the length of the second principal axis of the ellipse;
- $\mathcal{A}$, the area of the ellipse, equal to $\pi ab$.

The dynamical evolution of the ellipse is modeled by

$$\begin{cases} x_c(k+1) = x_c(k) + \Delta t.v_c(k).\cos(\alpha_c(k)) \\ y_c(k+1) = y_c(k) + \Delta t.v_c(k).\sin(\alpha_c(k)) \\ v_c(k+1) = v_c(k) + \Delta t.u_v(k) \\ \alpha_c(k+1) = \alpha_c(k) + \Delta t.u_\alpha(k) \\ \chi_e(k+1) = \chi_e(k) + \Delta t.u_X(k) \\ a(k+1) = a(k) + \Delta t.u_a(k) \\ b(k+1) = b(k) + \Delta t.u_b(k) \end{cases} \tag{46}$$

with

$$\begin{aligned} &u_{v-} \le u_v \le u_{v+},\ u_{\alpha-} \le u_\alpha \le u_{\alpha+} \\ &u_{\chi-} \le u_\chi \le u_{\chi+},\ u_{a-} \le u_a \le u_{a+}, \\ &u_{b-} \le u_b \le u_{b+} \end{aligned} \tag{47}$$

where $x_c, y_c, \chi_e, a, b$ are the aforementioned parameters of the ellipse, $v_c$ is the speed of the center and $\alpha_c$ is its orientation (see figure 16). The control inputs $u_v$, $u_\alpha$ govern the movement of the center of the ellipse $\mathbf{p}_c$ by acting on its speed and angular velocity, while the control inputs $u_\chi$, $u_\alpha$; $u_b$ modify the characteristic matrix $\mathbf{M}$ (shape and orientation).

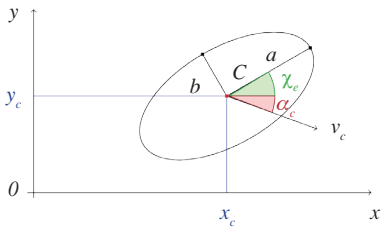



Fig. 16 - Ellipse parameterization

This dynamical model should be related to the dynamics of the UAVs so that it does not scatter the formation. The control inputs must thus be selected within a suitable range and suitable dynamics. This is yet a clear advantage over methods that abruptly modify the virtual structure and, as a result, do not take into account UAV constraints.

*Guidance law design*

The cost function $J^z$ associated with the motion and shape of the virtual structure is composed of terms dealing with the mission objectives and the constraints on the structure itself. The optimal control inputs at time $k$ should minimize the cost function $J^z$, such that

$$u_\chi^*, u_a^*, u_b^*, u_v^*, u_\alpha^* = \arg \min_{\substack{u_v, u_\alpha \\ u_\chi, u_a, u_b}} J^z \tag{48}$$

Where

$$J^z = J^{nav,direct} + J^{nav,horiz} + J^v + J^{abmin} + J^c \tag{49}$$

The components of $J^z$ are designed in such a way that

- $J^{nav,direct}$ drives the ellipse to a way-point and $J^{nav,horiz}$ constrains its speed to a desired value (costs defined in § "Costs used for all types of missions" and applied here to the ellipse center and velocity);

- $J^v$ keeps the ellipse area close to the initial one, $\mathcal{A}$;
- $J^{abmin}$ maintains $a$ and $b$ greater than a boundary value, so as to avoid the flattening of the ellipse along one of its axes;
- $J^c$ modifies the matrix $\mathbf{M}$ to avoid obstacles.

Costs related to ellipse constraints

$$J^v = w_v \sum_{t=1}^{H_p} \left\| \hat{a}(k+t)\hat{b}(k+1)\pi - \mathcal{A} \right\| \quad (50)$$

$$J^{abmin} = w_{abmin} \sum_{t=1}^{H_p} \left\{ f\left(\hat{a}(k+t), l_1, l_2\right) + f\left(\hat{b}(k+t), l_1, l_2\right) \right\} \quad (51)$$

where

$f(\hat{a}(k+t), l_1, l_2)$ is a continuous function, and $l_1$ and $l_2$ are user-defined parameters, such that

- $f$ takes the value 1 when $\hat{a}(k+t) \leq l_1$,
- $f$ takes the value 0 when $\hat{a}(k+t) \geq l_1 + l_2$,
- $f$ undergoes a continuous change between these two extreme values.

For example, an appropriate choice for $f$ is the function presented in figure 17, which is related to the one used in § "Applications".

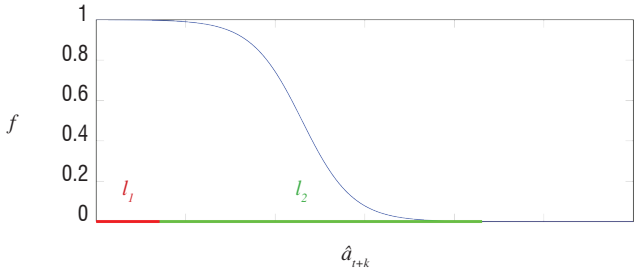



Fig. 17 - Function $f$

Costs related to obstacle avoidance

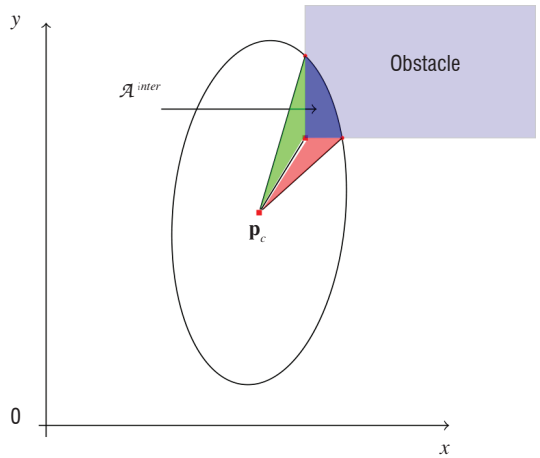



Fig. 18 - Ellipse-obstacle intersection

The structure should maneuver to avoid collision with obstacles in the environment. Assuming that the obstacles are described as convex surfaces (volumes in a 3D case), the intersection area between the virtual structure and the obstacles is computed to detect and quantify potential collisions (figure 18). Using this value as a penalization in the criterion makes it possible to find a path that minimizes this intersection and thus the risk of possible collision.

The collision avoidance term $J^c$ uses the intersection area $\mathcal{A}_l^{inter}(k+t)$ at time $k+t$ for each obstacle $l$ ($N^o$ being the number of obstacles in the neighborhood). The weight is chosen to give greater importance to the first prediction steps rather than the future ones.

$$J^c = w_c \sum_{t=1}^{H_p} \sum_{l=1}^{N^\circ} \frac{H_p - 1}{H_p} . \mathcal{A}_l^{inter}\left(k+t\right) \quad (52)$$

**UAV control**

The UAV control layer computes the motion of each vehicle so that it remains within the ellipse and avoids collision with the other vehicles. It thus has three goals:

- to attract the UAV inside the area;
- to allocate each UAV inside the area;
- to avoid collision between UAVs (using the cost defined in § "Safety cost").

This control is decentralized (each UAV determines its own control inputs), yet it uses the prediction of the future state of the virtual structure, which is available using the developments from § "Virtual structure control". MPC is used again, since allocation and collision avoidance may benefit from a prediction of the impact of control inputs on the future states of the vehicles.

The fleet is still composed of $N$ identical UAVs that are assumed to have instantaneous access to the state of all the other vehicles, which is $\mathbf{x}_i = [x_i, y_i, v_i, \chi_i]^T$ for the $i$-th vehicle.

For each UAV, the control inputs $u_i^v$ and $u_i^\omega$ are determined at each time $k$ in such a way that

$$u_i^{v*}, u_i^{\omega*} = \arg\min_{u_i^v, u_i^\omega} J_i \quad (53)$$

where for this application

$$J_i = J_i^{safe,veh} + J_i^u + J_i^t + J_i^e \quad (54)$$

The components of $J_i$ are designed in such a way that

- $J_i^{safe,veh}$ modifies the direction and the speed to avoid collision with other UAVs (§ "Safety cost"),
- $J_i^u$ minimizes the energy consumption in terms of control inputs (defined in § "Control cost"),
- $J_i^t$ drives the UAV within the area,
- $J_i^e$ keeps the speed and orientation of the UAV close to those of the center of the structure.

*Attraction and allocation of the UAVs within the structure*

The Mahalanobis distance [31] evaluates the norm between a point $\mathbf{p} = [x\ y]^T$ and the center $\mathbf{p}_c = [x_c\ y_c]^T$ of an ellipse, weighted by a function of the length of its main axis (see figure 19):

$$d_M(\mathbf{p}) = \sqrt{(\mathbf{p} - \mathbf{p}_c)^T \mathbf{M} (\mathbf{p} - \mathbf{p}_c)} \quad (55)$$

where $\mathbf{M}$ is the characteristic matrix of the ellipse.

The term $J_i^t$ is used to lead the UAVs within the virtual structure. The Mahalanobis distance is used to reflect the shape of the ellipse.

$$J_i^t = w_t \sum_{h=1}^{H_p} \frac{(H_p - h)}{H_p} g(i,h) \tag{56}$$

where the function g is defined as

$$g(i,h) = \begin{cases} d_M\left(\hat{\mathbf{p}}_i(k+h)\right) & \text{if } d_M\left(\hat{\mathbf{p}}_i(k+h)\right) > 1 \\ d_M\left(\hat{\mathbf{p}}_i(k+h)\right) - 1 & \text{if } d_M\left(\hat{\mathbf{p}}_i(k+h)\right) < 1 \end{cases} \tag{57}$$

The weight $(H_p - h)/H_p$ is meant to give more importance to the first predictions than the later ones. The function $g(i, h)$ is built on the basis of the Mahalanobis distance of the UAV position to the ellipse center. This function introduces a potential field that guides the UAV within the area. A discontinuity has been added, to make a stronger difference at the boundary of the virtual structure. A projection of function $g$ is provided in figure 20.

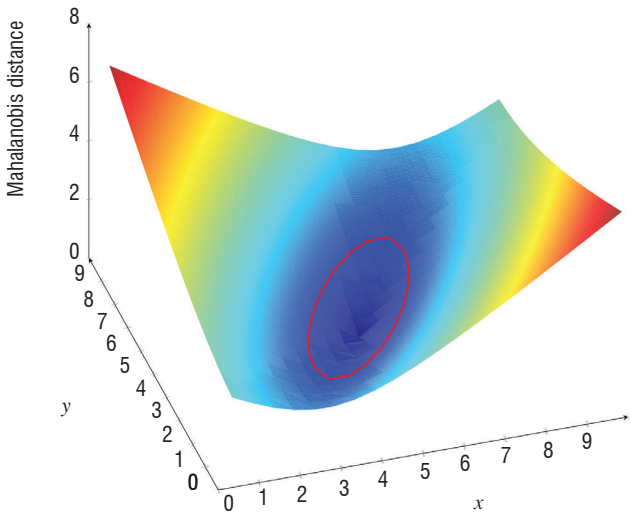



Fig. 19 - Mahalanobis distance to an ellipse (in red) over the position space

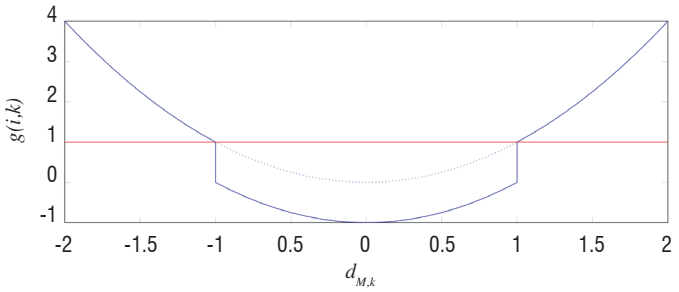



Fig. 20 - Shape of the function $g$ (2D projection)

*Formation consistency*

The cost $J_i^e$ is used to keep the speed and orientation of the UAV close to those of the ellipse $v_c$ and $\alpha_c$.

$$J_i^e = \sum_{t=1}^{H_p} w_n^t \left[ w_{ni}\left(v_i(k+t)\right) - v_c(k+t)\right)^2 + w_{n2}\left(\chi_i(k+t) - \alpha_c(k+t)\right)^2 \right] \tag{58}$$

The weights $w_n^t$ depend on the Mahalanobis distance between the UAV and the ellipse. When the distance is less than a value $l_3$, $w_n^t$ is equal to one and when it is greater than $l_4$, $w_n^t$ is equal to 0. The function $f$ (figure 17) is used again.

$$w_n^t = f\left(d_M\left(\mathbf{p}_i(t)\right), l_3, l_4\right) \tag{59}$$

**Simulation results**

A simple scenario has been defined to test the guidance law. The formation should reach a predefined way-point $\mathbf{p}_{target}$ starting from $\mathbf{p}_0$, with an initial orientation of the ellipse perpendicular to the motion direction. Two rectangular obstacles cross the trajectory of the formation. In order to avoid collision with these obstacles, the virtual structure must modify its shape so as to pass the obstacles safely. Only the deformation is considered here ($u_\chi = 0$), but an additional rotation of the structure could be handled similarly.

The initialization parameters of the simulation are given in table IX. Note that the virtual structure has a longer prediction horizon than the UAVs, since it holds more information on the final destination and the target.

| | | | |
|---|---|---|---|
| $w_{nav} = 10^{-2}$ | $w_v = 10^{-4}$ | $w_{ab} = 10$ | $w_{abmin} = 10^{-3}$ |
| $N = 8$ | $v_n = 4$ | $v_{min} = 2$ | $v_{max} = 6$ |
| $d\alpha_{max} = 0.3$ | $dt = 1$ | $d^v_{safe} = 6$ | $d^v_{des} = 6$ |
| $H_{p,uav} = 10$ | $H_{c,uav} = 5$ | $H_{p,ell} = 30$ | $H_{c,ell} = 5$ |
| $l_1 = 70$ | $l_2 = 90$ | $l_3 = 5$ | $l_4 = 10$ |
| $a_0 = 200$ | $b_0 = 100$ | $v_c = 4$ | $N° = 2$ |
| $\alpha_0 = \pi/2$ | $\mathbf{p}_0 = [100\ 0]$ | $\mathbf{p}_{target} = [2000\ 0]$ | |

Table IX - Simulation parameters

An example of gathering of the UAVs within the virtual structure is shown in figure 21.

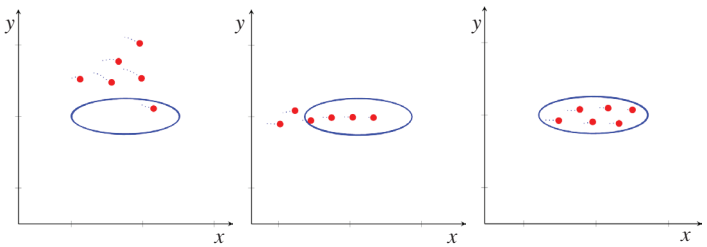



Fig. 21 - Gathering of UAVs within the ellipse structure

The complete scenario is illustrated by the sequence in figure 22. The ellipse modifies its shape accordingly when approaching the obstacles and no collision has been reported. The UAVs were initially in a vertical formation inside the ellipse. When the ellipse changes, the formation is modified to keep all of the UAVs within the structure. It can be seen that, since the range of the ellipse control inputs has been chosen to cope with the UAV dynamics, the vehicles have sufficient time to remain within the virtual structure when it is modified. The area of the ellipse is also kept close to its initial value. Figure 23 shows the values of the control inputs $u_a$ and $u_b$ that govern the deformation of the structure over time. These input values modify the length of the two principal axes simultaneously and almost symmetrically, to cope with the area constraint.

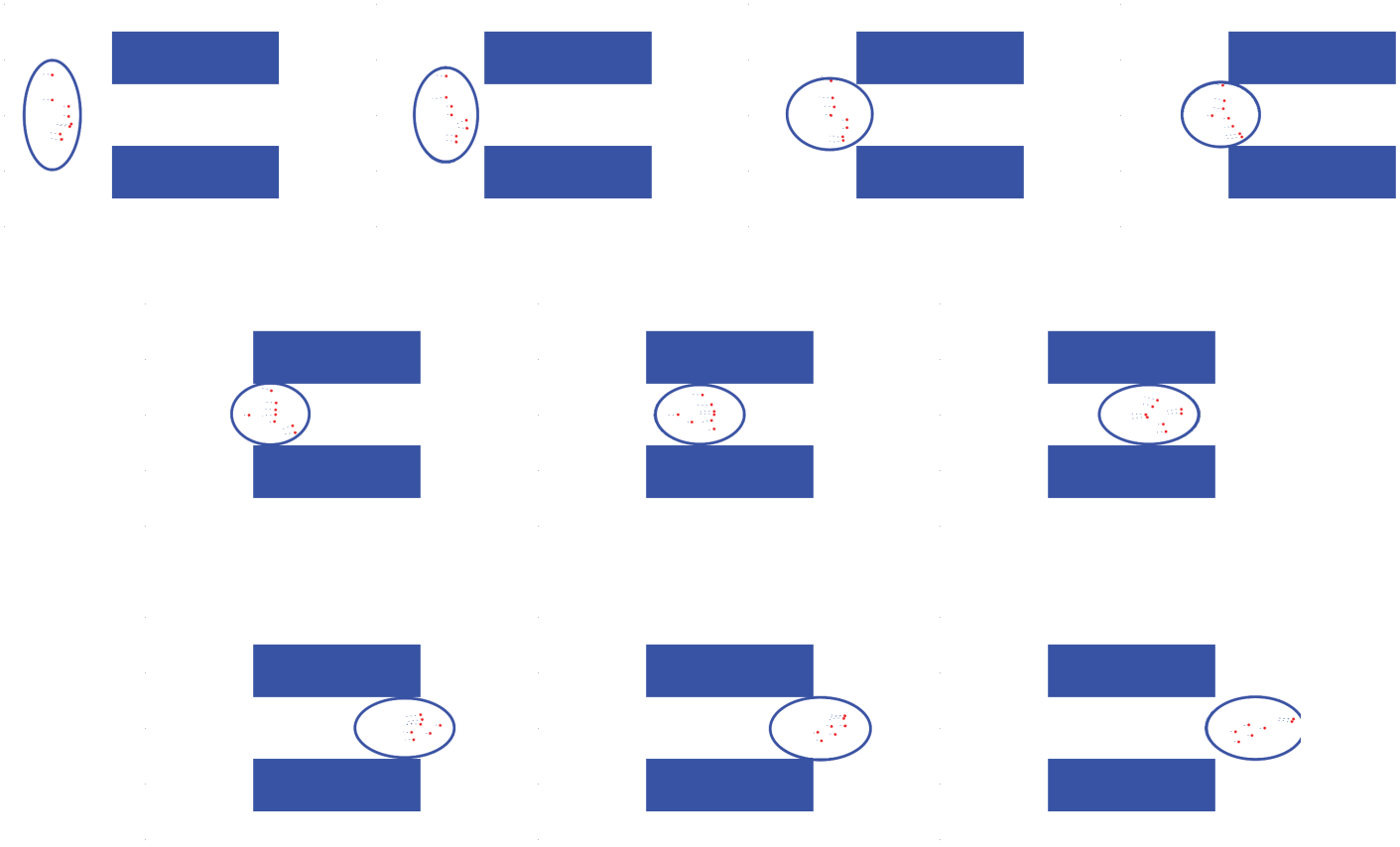


Fig. 22 - Obstacle avoidance by deformation

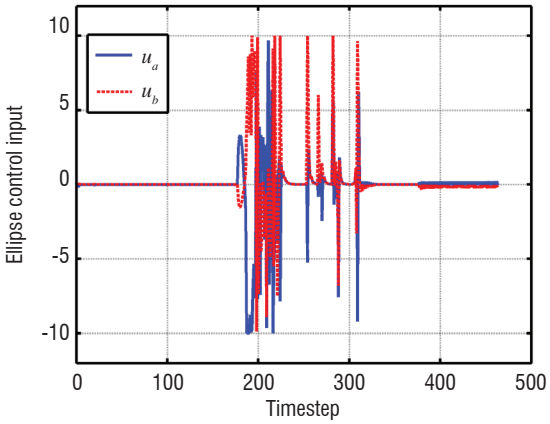



Fig. 23 - Control inputs for ellipse deformation

The proposed method thus makes it possible to maintain the UAVs within an elliptical virtual structure with collision avoidance, using the two-layer guidance law. The higher layer modifies the characteristics of the virtual structure with only knowledge of the obstacles and target, while the lower layer modifies the formation and distribution of the UAVs in a decentralized way, based only on the knowledge of the actions from the upper layer. Other shapes for the virtual structure could be taken into account within this guidance scheme by modifying the dynamical model of the structure and the criteria that govern the shape modification.

## Conclusions and perspectives

In this paper, the design of distributed cooperative control laws for a fleet of autonomous vehicles has been presented using Model Predictive Control. This approach proves very flexible for taking into account mission objectives and safety and reliability constraints. The use of an 'any-time' optimization procedure guarantees that a control value will be obtained in a given amount of time, depending on the computational ability of the vehicles. Future developments include detection and rejection of outlying data, definition of suitable observers taking the cooperativeness of the vehicles into account and demonstration of robustness properties of the resulting control laws ■

**Acknowledgements**

The authors would like to thank Dominique Beauvois, Didier Dumur, Tomasz Gorecki, Guillaume Broussin, Mathieu Touchard and Mathieu Balesdent who contributed to this research program.

**References**


[1] R. OLFATI-SABER - *Flocking for Multi-agent Dynamic Systems: Algorithms and Theory.* IEEE Transactions on Automatic Control, vol. 51, no. 3, pp. 401–420, 2006.

[2] A. JADBABAIE, J. LIN, A. S. MORSE - *Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules.* IEEE Transactions on Automatic Control, vol. 48, no. 6, pp. 988–1001, 2003.

[3] J. A. FAX, R. M. MURRAY - *Information Flow and Cooperative Control of Vehicle Formations.* IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1465–1476, 2004.

[4] D. GUNDUZ AND E. ERKIP - *Opportunistic Cooperation by Dynamic Resource Allocation.* IEEE Transactions on Wireless Communications, vol. 6, no. 4, pp. 1446–1454, 2007.

[5] T. VICSEK, A. CZIRÓK, E. BEN-JACOB, I. COHEN, O. SHOCHET - *Novel Type of Phase Transition in a System of Self-driven Particles.* Physical Review Letters, vol. 75, no. 6, p. 1226, 1995.

[6] B. GERKEY, M. MATARIC - *Sold!: Auction Methods for Multirobot Coordination.* IEEE Transactions on Robotics and Automation, vol. 18, no. 5, pp. 758–768, 2002.

[7] Y. GUO, L. E. PARKER - *A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots.* IEEE International Conference on Robotics and Automation, vol. 3, 2002, pp. 2612–2619.

[8] M. JUN, R. D'ANDREA - *Path Planning for Unmanned Aerial Vehicles in Uncertain and Adversarial Environments.* Cooperative Control: Models, Applications and Algorithms, Springer, 2003.

[9] W. DUNBAR, R. MURRAY - *Distributed Receding Horizon Control for Multi-vehicle Formation Stabilization.* Automatica, vol. 42, pp. 549–558, 2006.

[10] R. OLFATI-SABER, W. DUNBAR, R. MURRAY - *Cooperative Control of Multi-vehicle Systems Using Cost Graphs and Optimization.* Proceedings of the American Control Conference, Denver, Colorado, USA, vol. 3, 2003, pp. 2217–2222.

[11] W. DUNBAR, R. MURRAY - *Receding Horizon Control of Multi-vehicle Formations: A Distributed Implementation.* Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Bahamas, vol. 2, 2004, pp. 1995–2002.

[12] R. SCATTOLINi - *Architectures for Distributed and Hierarchical Model Predictive Control–a Review.* Journal of Process Control, vol. 19, no. 5, pp. 723–731, 2009.

[13] A. RICHARDS, J. P. HOW - *Robust Distributed Model Predictive Control.* International Journal of Control, vol. 80, no. 9, pp. 1517–1531, 2007.

[14] Y. ROCHEFORT, S. BERTRAND, H. PIET-LAHANIER, D. BEAUVOIS, D. DUMUR - *Cooperative Nonlinear Model Predictive Control for Flocks of Vehicules.* Proceedings of the IFAC Workshop on Embedded Guidance, Navigation and Control in Aerospace, Bangalore, India, 2012.

[15] E. HANSEN - *Global Optimization Using Interval Analysis*. CRC Press, 2004.

[16] P. POTOCNIK, I. GRABEC - *Nonlinear Model Predictive Control of a Cutting process.* Neurocomputing, vol. 43, no. 1-4, pp. 107–126, 2002.

[17] R. A. BARTLETT, A. WACHTER, L. BIEGLER - *Active Set vs. Interior Point Strategies for Model Predictive Control.* Proceedings of the 2000 American Control Conference, vol. 6, 2000, pp. 4229–4233.

[18] F. MARTINSEN, L. T. BIEGLER, B. A. FOSS - *A New Optimization Algorithm with Application to Nonlinear MPC*. Journal of Process Control, vol. 14, pp. 853–865, 2004.

[19] M. DIEHL, H. G. BOCK, J. P. SCHLODER, R. FINDEISEN, Z. NAGY, F. ALLGOWER - *Real-time Optimization and Nonlinear Model Predictive Control of Processes Governed by Differential-algebraic Equations.* Journal of Process Control, vol. 12, pp. 577–585, 2002.

[20] A. CHEMORI, M. ALAMIR - *Multi-step Limit Cycle Generation for Rabbit's Walking Based on a Nonlinear low Dimensional Predictive Control Scheme.* International Journal of Mechatronics, vol. 16, no. 5, pp. 259–277, 2006.

[21] M. ALAMIR, A. MURILO - *Swing-up and Stabilization of a Twin-pendulum under State and Control Constraints by Fast NMPC Scheme*. Automatica, vol. 44, pp. 1319–1324, 2008.

[22] E. W. FREW - *Receding Horizon Control Using Random Search for uav Navigation with Passive, Non-cooperative Sensing*. University of Colorado at Boulder, Tech. Rep., 2006.

[23] Y. ROCHEFORT - *Méthodes pour le guidage coopératif.* Ph.D. dissertation, Université Paris Sud - Supélec - Onera, 2013.

[24] T. GORECKI, H. PIET-LAHANIER, J. MARZAT, M. BALESDENT - *Cooperative Guidance of UAVs for Area Exploration with Final Target Allocation.* Proceedings of the 19th IFAC Symposium on Automatic Control in Aerospace, Würzburg, Germany, 2013.

[25] A. KAHN, J. MARZAT, H. PIET-LAHANIER - *Formation Flying Control Via Elliptical Virtual Structure*. Proceedings of the 10th IEEE International Conference on Networking, Sensing and Control, Evry, France, 2013, pp. 158–163.

[26] A. MOKHTARI, A. BENALLEGUE - *Dynamic Feedback Controller of Euler Angles and Wind Parameters Estimation for a Quadrotor Unmanned Aerial Vehicle.* IEEE International conference on robotics and automation, vol. 3, 2004, pp. 2359–2366.

[27] K. SINGH, K. FUJIMURA - *Map Making by Cooperating Mobile Robots.* IEEE International conference on robotics and automation, vol. 2, 1993, pp. 254–259.

[28] J. YUAN, Y. HUANG, T. TAO, F. SUN - *A Cooperative Approach for Multi-Robot Area Exploration*. IEEE International Conference on Intelligent Robots and Systems, 2010, pp. 1390–1395.

[29] J. MUNKRES - *Algorithms for the Assignment and Transportation Problems*. Journal of the Society of Industrial and Applied Mathematics, vol. 5, no. 1, pp. 32–38, March 1957.

[30] A. AHMADZADEH, G. BUCHMAN, P. CHENG, A. JADBABAIE, J. KELLER, V. KUMAR, G. PAPPAS - *Cooperative Control of UAVs for Search and Coverage*. Proceedings of the AUVSI conference on Unmanned Systems, Orlando, FL, USA, 2006.

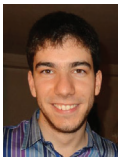
[31] R. D. MAESSCHALCK, D. JOUAN-RIMBAUD, D. MASSART - *The Mahalanobis Distance.* Chemometrics and Intelligent Laboratory Systems, vol. 50, no. 1, pp. 1–18, 2000.

## AUTHORS

**Sylvain Bertrand** received his Engineering degree in 2004 from the Ecole Centrale de Lille, and his PhD in 2007 from the University of Nice Sophia Antipolis. Since 2007 he has been working as a research engineer at Onera on optimization based control and estimation for dynamic systems, guidance of Unmanned Aerial Vehicles, orbit determination and performance evaluation of space systems.

**Julien Marzat** graduated as an engineer from ENSEM (INPL Nancy) in 2008. He completed his PhD Thesis from the University Paris Sud XI at the end of 2011. Since 2011, he has been working as a research engineer at Onera. His research interests include fault diagnosis and control of autonomous vehicles, as well as surrogate-based optimization for engineering design.

**Hélène Piet-Lahanier** is a Senior Scientist (MR2) at Onera. She graduated from SupAero (Toulouse) and obtained her PhD in Physics and her Habilitation à Diriger les Recherches (HDR), both from the University ParisXI, Orsay. Her research interests include modelling under uncertainty, bounded error estimation, cooperative guidance and autopilot design. She is a member of the IFAC technical committee on Aerospace.



**Arthur Kahn** received his Engineer degree from Polytech Paris UPMC and a master degree from University Paris VI both in 2012. He is currently preparing a Ph.D. degree in automation at ONERA, Palaiseau, France, in the System Design and Performance evaluation Department (DCPS). His fields of research are: control laws for autonomous vehicles, multi agent systems and robust estimation using sensor networks.

**Yohan Rochefort** graduated as an automation engineer from the Ecole National Supérieure de Physique de Strasbourg (ENSPS) in 2009. He obtained a PhD in automation from the Ecole Supérieure d'Electricité (SUPELEC) in 2013 on design of control laws for cooperative guidance of autonomous vehicles. He is currently working as a system engineer in the field of navigation.