

Cooperative guidance of UAVs for area exploration with final target allocation

Tomasz Gorecki, Hélène Piet-Lahanier, Julien Marzat, Mathieu Balesdent

▶ **To cite this version:**

Tomasz Gorecki, Hélène Piet-Lahanier, Julien Marzat, Mathieu Balesdent. Cooperative guidance of UAVs for area exploration with final target allocation. 19th IFAC Symposium on Automatic Control in Aerospace (ACA 2013), Sep 2013, Würzburg, Germany. pp.260-265. hal-00861973

HAL Id: hal-00861973
https://hal.science/hal-00861973

Submitted on 14 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cooperative guidance of UAVs for area exploration with final target allocation

Tomasz Gorecki [*,**] **Hélène Piet-Lahanier** [*] **Julien Marzat** [*]
Mathieu Balesdent [*]

[*] *ONERA – The French Aerospace Lab, F-91123 Palaiseau, France, {helene.piet-lahanier, julien.marzat, mathieu.balesdent}@onera.fr*
[**] *EPFL, Lausanne, Switzerland, tomasz.gorecki@epfl.ch*

Abstract: Fulfillment of complex missions, such as zone watching or multiple target tracking by an UAV can prove very demanding in terms of vehicle ability. A potential way of lessening these demands consists in splitting the initial task into several complementary subtasks. Each subtask can then be completed by one vehicle of a fleet whose cooperation must guarantee the satisfaction of the whole mission. In this paper, zone watching is defined as a cooperative problem where a number of autonomous vehicles must explore a wide area in a limited amount of time. In addition to zone coverage, the dynamical allocation of exit locations is considered. A model predictive control approach is adopted in which the requirements of the mission are specified as cost functions. Simulation results are presented to illustrate the behaviour of the fleet.

Keywords: autonomous vehicles, consensus, cooperative strategies, dynamical allocation, model predictive control

1. INTRODUCTION

Missions such as surveillance or multiple target tracking by autonomous vehicles are complex. In order to lower the cost induced by the development of such abilities on a single vehicle, research has focused on how such missions could be fulfilled using a fleet of cheaper and less sophisticated vehicles. However, to provide the same result, the vehicles must cooperate. It is required that:

- The combination of the individual actions fulfills the objectives of the global mission.
- Each trajectory ensures a secure flight for all vehicles in the fleet.

Cooperative control of vehicles is an active field of research: in Murray (2007), many applications in military and civil domains are surveyed, including formation flights (Jadbabaie et al. (2002)), cooperative attack (Rasmussen et al. (2004)), or wide area coverage (Ahmadzadeh et al. (2006a)).
The problem of area search or exploration has been studied in various contexts with different objectives: Burgard et al. (2005) address the collaborative mapping of an unknown area with a team of robots. Parker et al. (2000) propose a control architecture for the search of targets in a given zone. Maza and Ollero (2007) allocate search areas to UAVs by subdividing the mission field into smaller parts assigned according to the vehicle capabilities. Ahmadzadeh et al. (2006a) tackle cooperative surveillance by an UAV fleet by proposing a heuristics which aims at reducing the time any area remains unwatched. Ahmadzadeh et al. (2006b) consider a limited time exploration scenario of a mission field by a fleet of vehicles with constrained entry and exit points: the problem is solved offline by constructing an admissible flight plan for each vehicle, ensuring collision and obstacle avoidance through a tree search in a discretised trajectory search space. Choset (2001) reviews algorithms for exploring an unknown zone by a group of vehicles while ensuring total coverage.
This paper adresses a more complex exploration setup. The main objective of the mission is to explore the unknown area during a fixed mission duration. In addition, cooperative target allocation is considered as the vehicles have to rejoin constrained exit locations at the end of the mission. The problem considered is to define vehicle cooperative trajectories so that:

- the cumulated explored area is maximised,
- the mission duration is fixed in advance,
- the vehicles must reach a safe position at the end of the mission,
- each vehicle defines its own trajectory online.

The trajectories are individually chosen using model predictive control (MPC) based on exploration and safety costs. A method for the target allocation derived from the MPC framework is presented. It consists in approximating the cost to go to an exit based on the MPC optimal cost to evaluate the suitability of each exit point. The tractability of the proposed method allows dynamic reallocations, while limiting the amount of information shared between the vehicles, and therefore preserves the decentralised scheme.

2. PROBLEM DESCRIPTION

2.1 Mission definition

The scenario consists in defining cooperative UAV trajectories that maximise a zone coverage in a given time

denoted by T_{end} while ensuring collision avoidance and restrictions on the final positions of the vehicles. Each vehicle has to reach one of a set of predefined exit zones at the end of the mission. These exit points represent shelters or filling stations. Each of these zones can shelter a limited number of vehicles, denoted n_{max}. Therefore, the exit point allocation requires to obtain a consensus between the vehicles to insure that each of them reaches an exit point before the end of the mission and the sheltering capacities of the exit point are not exceeded.

The mission objectives are thus:

- Area coverage (Section 4)
- Allocation and reaching of the exit points (Section 5)

while the constraints are:

- Collision avoidance (Section 3)
- Limited mission time (Section 5)
- Limited number of vehicles at a given exit (Section 5)

2.2 UAV model

The n_v UAVs are assumed to be identical. The trajectories are assumed to stay in an horizontal plane and the UAVs are pointwise. The state and control vectors of each vehicle i are defined as:

$$X_i = \begin{pmatrix} x_i \\ y_i \\ v_i \\ \Psi_i \\ \omega_i \end{pmatrix} \text{ and } U_i = \begin{pmatrix} u_i^\omega \\ u_i^v \end{pmatrix} \tag{1}$$

with x, y the vehicle position, v its speed amplitude, Ψ its direction and ω its rotation speed. The model kinematics are:

$$\begin{cases} \dot{x_i} &= v_i \cos \Psi_i \\ \dot{y_i} &= v_i \sin \Psi_i \\ \dot{v_i} &= u_i^v \\ \dot{\Psi}_i &= \omega_i \\ \dot{\omega_i} &= u_i^\omega \end{cases} \tag{2}$$

(u_i^v, u_i^ω) are the longitudinal and rotational accelerations. This model, though simplified, has already been used in the literature: Richards and How (2007); Rochefort et al. (2012). Further research should include more realistic dynamics. The constraints on the states and control inputs are:

$$\begin{array}{cc} v_{min} \leq v_i \leq v_{max} & -\omega_{max} \leq \omega_i \leq \omega_{max} \\ -\Delta v_{max} \leq u_i^v \leq \Delta v_{max} & -\Delta\omega_{max} \leq u_i^\omega \leq \Delta\omega_{max} \end{array} \tag{3}$$

We summarize the dynamics and the constraints as:

$$\dot{X}_i = f(X_i, U_i) \text{ and } (X_i, U_i) \in \mathcal{X}_i \times \mathcal{U}_i \tag{4}$$

The superscript t denotes time.

Communication delays and ranges are not considered here, all the UAVs are assumed to have access to the state of every vehicle instantly.

2.3 Control framework

MPC has been widely used for UAVs in different contexts. UAV flocking and formation flight has been treated in Dunbar and Murray (2006). Distributed MPC is studied in Richards and How (2007); Rochefort et al. (2012). In distributed MPC (Scattolini (2009)), each vehicle computes its control inputs at each timestep as a solution of an optimisation problem over the future predicted trajectory. For tractability reasons, finite prediction and control horizons, respectively denoted H_p and H_c are used.

The future control inputs and the resulting state trajectories are denoted:

$$\mathbf{U_i} = ((U_i^{t_0})^T, (U_i^{t_0+1})^T, ..., (U_i^{t_0+H_c-1})^T)^T$$
$$\mathbf{X_i} = ((X_i^{t_0+1})^T, (X_i^{t_0+2})^T, ..., (X_i^{t_0+H_p})^T)^T$$

When $H_c < H_p$, we assume that the control inputs are 0 after H_c steps. Once the optimal input sequences $\mathbf{U_i}^\star$ have been computed, each vehicle communicates its predicted trajectory to the rest of the fleet and applies the first entry $U_i^\star(t_0)$. The optimisation problems at time t_0 take the form:

$$\begin{gathered} \text{minimise } J_i(\mathbf{U_i}, \mathbf{X_i}) \\ \text{over } \mathbf{U_i} \in (\mathcal{U}_i)^{H_c} \\ \text{subject to} \\ \forall t \in [t_0+1; t_0+H_p], X_i^t \in \mathcal{X}_i \end{gathered}$$

J_i is the cost function for vehicle i. The constraints coupling the vehicles, such as collision avoidance, are included by means of a penalty factor in the cost function. At the next timestep, each vehicle solves its optimisation considering that the other vehicles will follow their predicted trajectories.

The cost function J_i is composed of a sum of terms reflecting the objectives of the mission. In the following sections, these terms are detailed. Selection of the weights in the cost functions is discussed in Section 6.1.

3. COLLISION AVOIDANCE

The main constraints on the vehicles relative trajectories are to avoid collision. These constraints are non-convex and thus difficult to enforce at all time. Therefore, we adopted a soft version of these constraints by means of a cost function. The difficulty is brought in the cost function which is nonconvex, but the optimisation is remains feasible.

$$J_i^{coll} = W^{coll} \sum_{t=t_0+1}^{t_0+H_p} \sum_{\substack{j=1 \\ j\neq i}}^{n_v} \frac{1 + \tanh\left((d_{ij}(t) - \beta_a)\alpha_a\right)}{2} \tag{5}$$

where $d_{ij}(t) = \|p_i(t) - p_j(t)\|$ denotes the distance between vehicles i and j, and $p_i = (x_i, y_i)^T$. α_a and β_a respectively parameterize the width of the interval of fast variation of the hyperbolic tangent and its inflexion point. Note that the hyperbolic tangent function is nearly constant out of

the domain $[d_{danger}, d_{safe}]$ which is related to α_a and β_a by:

$$\alpha_a = \frac{6}{d_{safe} - d_{danger}} \qquad \beta_a = \frac{1}{2(d_{danger} + d_{safe})}$$

With this choice, the cost variation is less than 5% of its maximal value in the range $[d_{safe}, +\infty[$. For implementation, the penalty function is set to 0 for $d_{ij} > d_{safe}$, *i.e.* the vehicles do not "see" each other above this distance.

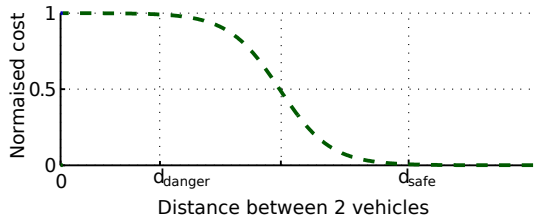


Fig. 1. Penalty function for collision avoidance

4. ZONE COVERAGE

A cost function that reflects the gain in terms of map exploration for a potential trajectory is defined. Each vehicle is assumed to have a seeker capability described by a function f_{expl} of the relative position of the observed point and the vehicle.

4.1 Exploration modeling

The previously explored area is:

$$\Omega = \bigcup_{\substack{t=1,\dots,t_0 \\ i=1,\dots,n_v}} \mathcal{D}_i^t \tag{6}$$

with $\mathcal{D}_i^t$ the sensing footprint of the vehicle i at time t. As 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 $G_{k,l}$ ranges between 0 when no vehicle has explored this square and 1 when it has been entirely explored. 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 (k, l), the exploration level is updated in the following fashion:

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

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 $d > r_{max}$. We consider here:

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

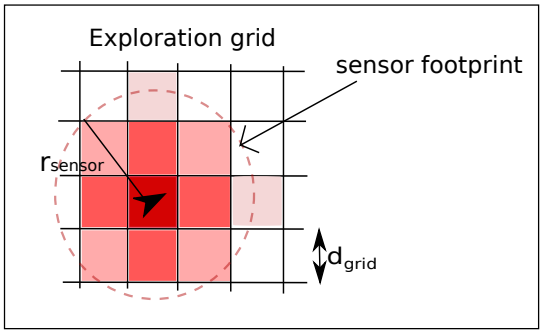


Fig. 2. Illustration of exploration cost: colours reflect the exploration level

4.2 Cost function description

The cost function rewards trajectories increasing the global level of exploration of the map. It is defined as:

$$\begin{aligned} J_i^{expl}(\mathbf{X_i}^{t_0}) &= -W^{expl} \sum_{k,l} (G_{k,l}^{t_0+H_p} - G_{k,l}^{t_0}) \\ &= -W^{expl} \mathbf{1}^T . (G^{t_0+H_p} - G^{t_0}) . \mathbf{1} \end{aligned} \tag{7}$$

where $G^{t_0+H_p}$ is the predicted exploration map associated with trajectory $\mathbf{X_i}$ and $\mathbf{1}$ the vector whose components are all 1. This cost function represents the total increase of the global exploration level resulting from a predicted trajectory. As the vehicles share information, they are penalised when flying in already explored zones.

5. EXIT POINT ALLOCATION AND TRACKING

We first describe how a vehicle joins a given exit point and present how this point is selected.

5.1 Exit point rejoining

The cost for reaching a point takes the following form (Rochefort et al. (2012))

$$\begin{aligned} J_i^m = \sum_{t=t_0}^{t_0+H_p} & [\varphi_i^{mv}(X_i^t, U_i^t) + \varphi_i^{m\omega}(X_i^t, U_i^t) \\ & + \varphi_i^{mt}(X_i^t, U_i^t, t)] + \Phi_i^{mf}(X_i^{t_0+H_p})] \end{aligned} \tag{8}$$

with

$$\begin{aligned} \varphi_i^{mv}(X_i^t, U_i^t) &= W^{mv}(v_i^t - v_{opt})^2 \\ \varphi_i^{m\omega}(X_i^t, U_i^t) &= W^{m\omega}(\omega_i^t)^2 \\ \varphi_i^{mt}(X_i^t, U_i^t, t) &= \frac{1}{t} W^{mt} \|p_i^t - \hat{p}_i^t\| \\ \Phi_i^{mf}(X_i^{t_0+H_p}) &= W^{mf} D(p_i^{t_0+H_p} - \hat{B}_i^{t_0+H_p}) \end{aligned} \tag{9}$$

φ_i^{mv} penalises the difference between the actual speed and the desired speed v_{opt}, $\varphi_i^{m\omega}$ favours straight lines over curved trajectories, φ_i^{mt} penalises the distance of the predicted trajectory to a virtual best-case trajectory $\hat{p}_i$ which is a straight line towards the exit point at speed v_{opt} and Φ_i^{mf} encourages the vehicle to get closer to the exit point by penalising the distance to a reference ball $\hat{B}_i$ around the exit point illustrated in Figure 3.

5.2 Exit point allocation

Two cases are studied. In the first one, the number of vehicles n_v is identical to the number of exits n_t and a

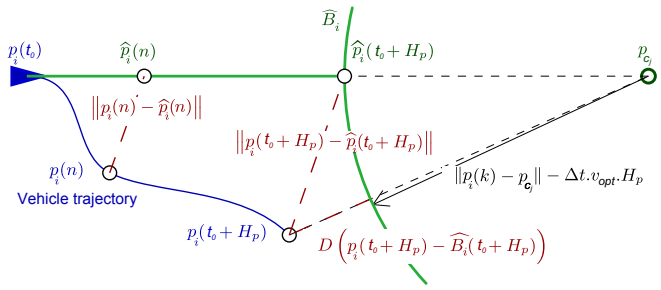


Fig. 3. Reference trajectory $\hat{p}_i$ and reference ball $\hat{B}_i$

given exit point can shelter at 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 $\mathbf{n_t = n_v}$ The aim is to define a cost which balances the distance to the exit and the cost in the control inputs (in other words, favours vehicle which go straight to the exit compared to ones which change direction). 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 has to discriminate efficiently between different vehicles aiming for the same exit.

For each pair of vehicle and exit, the cost function:

$$J^{af}_{i,j}({\mathbf{U_i}}^{t_0}, {\mathbf{X_i}}^{t_0}) = \sum_{t=t_0}^{t_0+H_p-1} W^d \|p_i^t - p_{c_j}\|^2 + W^u \|U_i\|^2 \quad (10)$$

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

$$r_{ij} = \min_{\mathbf{U_i}} J^{af}_{i,j}(\mathbf{U_i}, \mathbf{X_i})$$

The optimal assignment is obtained by the hungarian algorithm (Munkres (1957)).

Case $\mathbf{n_t < n_v}$ In this case, at most n_{max} vehicles can go to the same exit point at the end of the mission. The $(n_v \times n_t)$ matrix R is built. A heuristics is used to find a good admissible assignment: each vehicle forms a list of wishes based on his cost evaluations. These costs are centralised and it is checked whether no more than n_{max} vehicles aim for each exit. In case of conflict, the admissible alternative exits are considered. The minimising costs among these are chosen for each conflicting vehicle, one after the other. In these two cases, the construction of the cost matrix is decentralised but information has to be centralised to perform the actual assignment.

5.3 Dynamic reallocation of exit points

We have described a situation where the vehicles 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 time 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

k^{mt}	$\frac{1}{2.H_p.v_{max}}$	k^{mf}	$\frac{1}{2.H_p.v_{max}}$
k^{coll}	$\frac{2}{H_p}$	k^{expl}	$\left(\frac{2r_{sensor}.v_{opt}.H_p}{d_{grid}}\right)^{-1}$
k^{u}	$\frac{1}{(H_c.v_{max}.\omega_{max})^2}$	k^{d}	$\frac{1}{\overline{dist}^2}$

Table 1. Renormalisation coefficients

of the vehicles 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 optimisation assigns an exit to a vehicle that cannot reach it 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:

$$s_{ij} = \begin{cases} 0 & \text{if } T \geq T_{safe} \\ \left(\dfrac{T_{safe} - T}{T_{safe} - T_{danger}}\right)^2 & \text{if } T < T_{safe} \end{cases} \quad (11)$$

where T is the remaining time, $T_{danger} = \frac{\|p_i - p_{c_j}\|}{v_{opt}}$ and $T_{safe} = f_{safe}.T_{danger} + T_{margin}$, c_j the position of the exit. T_{margin} and f_{safe} are predefined parameters. The matrix used for the global assignment is $R' = (r'_{ij})_{\substack{i=1,\dots,n_v \\ j=1,\dots,n_t}}$ with $r'_{ij} = r_{ij} + s_{ij}$. The continuous variation of the penalty prevents vehicles from choosing unreachable exit points provided 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 performances because only estimates of the costs to go are required in order to choose a reasonable assignment. The linear approximation and the constraints translation is taken from (Richards and How (2007)). The reallocation does not need to be repeated at each timestep, but only from time to time.

6. GLOBAL PENALTY FUNCTION

The choice of optimal control entries should take into account the three main aspects previously described: collision avoidance, map exploration, and exit point assignment. Therefore, the cost function is:

$$J_i = J_i^{coll} + J_i^{expl} + J_i^{m} \quad (12)$$

6.1 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 normalisation coefficient and $w^\bullet$ a weighting coefficient. The $k^\bullet$ (Table 1) coefficients are chosen so that without weighting, the worst case cost would be around 1.

Note that k^d is chosen so that exit allocation can be fairly performed between the vehicles:

$$\overline{dist} = \frac{1}{n_t.n_v} \sum_{i=1}^{n_v} \sum_{j=1}^{n_t} \|p_i - p_{c_j}\| \quad (13)$$

is the average distance between vehicles and targets.

v_{min}	$0.3\ m.s^{-1}$	v_{opt}	$0.7\ m.s^{-1}$	v_{max}	$1.0\ m.s^{-1}$
$\Delta\omega_{max}$	$0.1\ rd.s^{-1}$	Δv_{max}	$0.\ 1m.s^{-1}$	ω_{max}	$0.5\ rd.s^{-1}$
d_{danger}	$4\ m$	d_{safe}	$8\ m$	w^{coll}	5
w^{mf}	1	w^{mt}	2	w^{expl}	2
w^{u}	0.5	w^{mv}	0.3	$w^{m\omega}$	0.2
w^{d}	2	f_{safe}	1.1	C_{min}	0.2
H_c	$2\ s$	H_p	$21\ s$	r_{max}	$5\ m$
T_{margin}	$15\ s$	d_{grid}	$2.5\ m$		

Table 2. Simulation parameters

6.2 Dynamic weighting of the functions

As the total time allocated for the mission is known, it is preferable to join 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 inspired by Ahmadzadeh et al. (2006a) is adopted: the exploration of the map is initially favoured in the cost function, whereas exit points progressively take more importance in the cost function as the remaining time decreases. This translates in the algorithms by means of balancing coefficients C^{expl}, C^{mt} and C^{mf} which multiply W^{expl}, W^{mt} et W^{mf} as reported in Algorithm 1.

Algorithm 1 Calculation of the weighting coefficients

(1) Compute $d = dist(p_i, c_i)$ distance between vehicle i and its exit point.

(2) Compute $T_{danger} = \frac{d}{v_{opt}}$ 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:

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

$$C^{mt} = C^{mf} = \begin{cases} 1 & \text{if } T \leq T_{danger} \\ \frac{C_{min}(T - T_{danger}) + T_{safe} - T}{T_{safe} - T - danger} & \text{if } T_{danger} < T \leq T_{safe} \\ C_{min} & \text{if } T > T_{safe} \end{cases}$$

7. SIMULATION RESULTS

This section presents simulation results illustrating the procedure. Simulation parameters are grouped in Table 2. The different requirements of the mission are first individually illustrated and quantitative simulation results are then given.

7.1 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 current time. The circles denote the danger zones around the vehicles. Figure 4 shows that vehicles can reach agreements even in complex situations to cross ways without endangering themselves or the other vehicles.

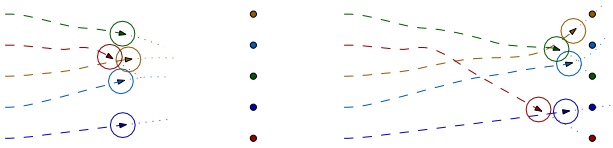
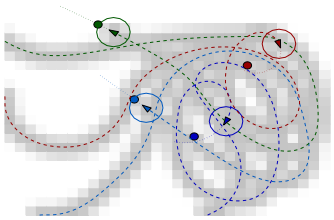


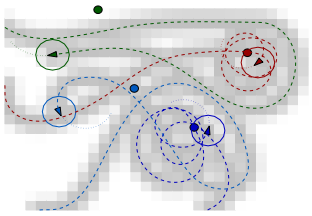
Fig. 4. Illustration of the collision avoidance

7.2 Map exploration and exit assignment

Map exploration and exit assignment are illustrated with a 4 vehicle scenario presented in Figure 5. Exits are chosen randomly for each vehicle and no reallocation is allowed. It compares the behaviour 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.



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



(b) Weighting factors are constant

Fig. 5. Comparison of different exploration strategies

7.3 Dynamic reallocation

Dynamic reassignment is illustrated in Figure 6. The current assignment in the figures is depicted by matching colours. The mission state is displayed at different instants of the mission. In this particular instance, the vehicle beginning in the top left corner conserves his initial assignment during the mission, whereas the two others do not. One of them first changes his exit whereas the last one also changes his plan later on.

7.4 Performances

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

- No exploration is considered: each vehicle chooses an exit and joins it as soon as possible. (A)

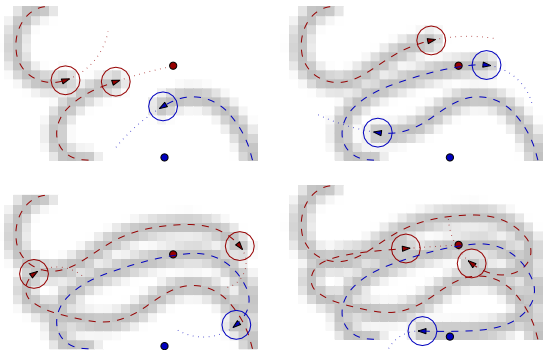


Fig. 6. Online reassignment of the exit points: colours of the vehicles correspond to the assigned target

	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 3. Simulation results (standard deviations are given between brackets)

- 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 term. (C)
- Configuration is identical to (C) but reassignment is authorised. (D)

In each mission, a 78mx78m 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 3. Expl(%) is the portion of the map which 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 amount of mission during which a dangerous situation has occured, that is when 2 vehicles come closer than a distance of d_{danger} 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 even more the efficiency of the exploration, as expected. However, it also increases the number of dangerous situations: adding the exploration cost increases the chances for several vehicles to come in the same zone and therefore increases the collision risks. What is more, dynamical reallocation reduces these risks significantly, while preserving the exploration efficiency.

8. CONCLUSIONS AND PERSPECTIVES

A cooperative guidance approach has been presented for wide zone surveillance by a fleet of UAVs. The exploration is time-constrained and each vehicle must reach a safe exit platform at the end of the mission. Platforms can shelter a limited number of vehicles, so a dynamic allocation method is proposed. During the mission, the cooperative guidance laws based on MPC dynamically balance exploration and exit assignment while avoiding collisions. The results presented show that a global complex mission can be satisfactorily fulfilled by use of a short-sighted and distributed architecture of control. This method renders the problem tractable and allows to choose the actions to be taken online. Further developments include 3D representation, further investigation of the optimisation methods and more realistic UAV representations including communication delays and state estimation.

REFERENCES

Ahmadzadeh, A., Buchman, G., Cheng, P., Jadbabaie, A., Keller, J., Kumar, V., and Pappas, G. (2006a). Cooperative control of uavs for search and coverage. In *Proceedings of the AUVSI conference on unmanned systems, Orlando, USA*.

Ahmadzadeh, A., Keller, J., Jadbabaie, A., and Kumar, V. (2006b). Multi-UAV cooperative surveillance with spatio-temporal specifications. In *Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, USA*.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. (2005). Coordinated Multi-Robot Exploration. *IEEE Transactions on Robotics*, 21(3), 376–386.

Choset, H. (2001). Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31, 113–126.

Dunbar, W. and Murray, R. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42, 549–558.

Jadbabaie, A., Lin, J., and Morse, A.S. (2002). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48, 988–1001.

Maza, I. and Ollero, A. (2007). Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. *Distributed Autonomous Robotic Systems*, 6, 221–230.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1), 32–38.

Murray, R. (2007). Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5), 571–583.

Parker, L. et al. (2000). Current state of the art in distributed autonomous mobile robotics. *Distributed Autonomous Robotic Systems*, 4, 3–12.

Rasmussen, S., Shima, T., Mitchell, J., Sparks, A., and Chandler, P. (2004). State-space search for improved autonomous uavs assignment algorithm. In *Proceedings of the 43rd IEEE Conference on Decision and Control, Kingston, Canada*, volume 3, 2911–2916.

Richards, A. and How, J.P. (2007). Robust distributed model predictive control. *International Journal of Control*, 80(9), 1517–1531.

Rochefort, Y., Bertrand, S., Piet-Lahanier, H., Beauvois, D., and Dumur, D. (2012). Cooperative nonlinear model predictive control for flocks of vehicles. In *Proceedings of the IFAC Embedded Guidance Navigation and Control in Aerospace Conference, Bangalore, India.*

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control–a review. *Journal of Process Control*, 19(5), 723–731.