Control Engineering Practice 31 (2014) 204–217

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Model predictive control of cooperative vehicles using systematic search approach

Y. Rochefort [a,b,*], H. Piet-Lahanier [a], S. Bertrand [a], D. Beauvois [b], D. Dumur [b]

[a] ONERA - The French Aerospace Lab, Palaiseau, France
[b] SUPELEC Systems Sciences E3S, Gif-Sur-Yvette, France

ARTICLE INFO

Article history:
Received 15 July 2012
Accepted 5 January 2014
Available online 11 February 2014

Keywords:
Cooperative model predictive control
Distributed control
Multi-vehicle systems
Unmanned aerial vehicle
Obstacle avoidance

ABSTRACT

This paper describes the guidance law design of a group of autonomous cooperative vehicles using model predictive control. The developed control strategy allows one to find a feasible near optimal control sequence with a short and constant computation delay in all situations. The control strategy takes other vehicles predicted positions into account for cooperation purpose. Numerical simulations are provided where a group of quadrotors must reach several way-points while avoiding obstacles and collisions inside the group. Results obtained using a realistic model of small quadrotors show that the approach could be usable in practice.

1. Introduction

In recent years, the interest for autonomous vehicles has been steadily growing. This type of vehicles finds its place in all kinds of missions, civilian (search and rescue, transportation, foraging, mobile grid of sensors, repairing) or military (recognition, surveillance, combat). Autonomous vehicles thrive in missions that are too dull, dirty or dangerous for human beings.

Using several cooperative vehicles stems from the idea that the global performances obtained would overcome those of a single vehicle. The first point is that the mission could be distributed among the vehicles, lessening the requirements in terms of technological burden. Another argument in favour of cooperation is enhanced robustness. If a failure occurs on one of the vehicles, it remains possible to (at least partially) re-affect the mission to the remaining flock. Another point is that with simpler vehicles, the probability of occurrence of a failure may be lower.

The control of a group of cooperative vehicles can be achieved by centralised (Wang, Yadav, & Balakrishnan, 2007) or distributed control (Rochefort, Piet-Lahanier, Bertrand, Beauvois, & Dumur, 2011; Siva & Maciejowski, 2011).

In a centralised scheme, actions of all vehicles are decided by a single element using all the available information. This allows one to elaborate the actions of the various vehicles all at once, easing their synchronisation and optimisation. This however requires potentially heavy computation to solve the unique centralised problem which may limit the number of vehicles that can be taken into account. It also demands a high level of robustness of the transmission links in order to keep track of all vehicles. Finally, the element responsible for the control computation could represent a single point of failure if no other element is able to take its place.

On the other hand, distributed control lessens the computational burden as each vehicle computes its own control input and therefore allows better scalability and robustness. Drawbacks are twofold: first, the available computational abilities of each vehicle may be a limiting factor for solving complex problems and second, information available about other vehicles can be inaccurate, incomplete or known with a delay.

Cooperative control law designs require to take into account the behaviour of a vehicle together with those of the others. In order to be more efficient, the control law should be based on both current and predicted states of all the vehicles. Hence, Model Predictive Control (MPC) seems a natural candidate for such laws. Indeed, in MPC the control input is computed by minimising a cost function which depends on predictions of the vehicles' behaviours over a finite time horizon.

Originally introduced to control systems with very slow dynamics like chemical plants or temperature regulation, the MPC strategy is now used to control systems with fast dynamics like Unmanned Aerial Vehicles. In Kim and Shim (2003) and Kim, Shim, and Sastry (2002), an MPC approach is used to control a six-degree-of-freedom nonlinear helicopter model. A gradient descent procedure is used to solve the optimisation problem, but closed loop stability issues are not addressed. In Bertrand, Piet-Lahanier, and Hamel (2007), the authors give a stability analysis of a contractive Non-linear MPC (NMPC) that allows an isolated vehicle to follow a predefined trajectory. In Frew

* Corresponding author.
E-mail addresses: yohan.rochefort@laposte.net (Y. Rochefort), helene.piet-lahanier@onera.fr (H. Piet-Lahanier), sylvain.bertrand@onera.fr (S. Bertrand), dominique.beauvois@supelec.fr (D. Beauvois), didier.dumur@supelec.fr (D. Dumur).

0967-0661/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.conengprac.2014.01.003

(2006), the author presents an MPC with a random search algorithm that allows a single autonomous flying vehicle to either explore its environment, track a predefined trajectory or join some way points in a clustered environment.

Current research on cooperative distributed MPC focuses mainly on the proof of convergence (Dunbar & Murray, 2004; Müller, Reble, & Allgöwer, 2011; Venkat, Rawlings, & Wright, 2005), robustness (Siva & Maciejowski, 2011), and formation flying (Olfati-Saber, Dunbar, & Murray, 2003) without addressing computation delay issues. For a single system with fast dynamics, real time feasibility of Non-linear MPC has been demonstrated, *e.g.* on the Caltech ducted fan (Dunbar, 2001) or a twin-pendulum (Alamir & Murilo, 2008). For large scale distributed systems, Scattolini (2009) presents a review of selected MPC approaches.

In this paper, a distributed cooperative control algorithm based on MPC for a group of autonomous flying vehicles is proposed. To allow cooperation, each vehicle communicates its predicted positions to the others, which are taken into account by the MPC approach. The MPC strategy also allows one to take into account constraints of the vehicle (actuator limitations, maximum velocities, etc.). In addition, the cost function can be designed to integrate the required performances of all the tasks linked with the mission.

Originally introduced in Rochefort, Piet-Lahanier, Bertrand, Beauvois, and Dumur (2012), this approach is extended in this paper to three-dimensional problems. An application to realistic quadrotor models is also described. Experimental results on ground mobile robots are also proposed to complete simulation results.

In Section 2, the problem is stated. Section 3 exposes the control strategy for the vehicles. Section 4 contains a short explanation of the MPC approach whereas the objective function is detailed in Section 5. Section 6 presents the proposed control sequence search procedure. The robustness of this procedure is evaluated in Section 7 and compared to a classical optimisation procedure. A short study of the influence of various parameters of the control search procedure is also provided. Experimental results on ground mobile robots are finally presented to complete simulation results. Conclusion and perspectives of works end this paper.

2. Problem statement

A system composed of N autonomous vehicles moving in a three dimensional space is considered. The model of the translational dynamics of a vehicle i is a discrete-time double integrator:

$$\mathbf{p}_i(k+1) = \mathbf{p}_i(k) + \Delta t.\mathbf{v}_i(k) \tag{1}$$

$$\mathbf{v}_i(k+1) = \mathbf{v}_i(k) + \Delta t.\mathbf{a}_{d,i}(k) \tag{2}$$

where Δt is the sampling time. The state $\mathbf{x}_i(k) = [\mathbf{p}_i(k)^t\ \mathbf{v}_i(k)^t]^t$ of a vehicle is composed of its position $\mathbf{p}_i(k) = [p_i^x(k)\ p_i^y(k)\ p_i^z(k)]^t$ and velocity $\mathbf{v}_i(k) = [v_i^x(k)\ v_i^y(k)\ v_i^z(k)]^t$. $\mathbf{p}_i$ and $\mathbf{v}_i$ are defined in a common inertial frame $\mathcal{F}$ whose horizontal axes are $x_\mathcal{F}$ and $y_\mathcal{F}$ and vertical axis $z_\mathcal{F}$ is directed downward as seen in Fig. 1. The control input of the vehicle i is composed of the desired accelerations $\mathbf{a}_{d,i}(k) = [a_{d,i}^x(k)\ a_{d,i}^y(k)\ a_{d,i}^z(k)]^t$ along the three axes of $\mathcal{F}$.

The vectors of horizontal velocity and desired acceleration are respectively denoted $\mathbf{v}_i^h = [v_i^x\ v_i^y]^t$ and $\mathbf{a}_{d,i}^h = [a_{d,i}^x\ a_{d,i}^y]^t$.

Vehicles' control inputs and states are constrained by practical limitations. In the simple model presented above, limitations are represented by constraints (3)–(6) and apply, at each time k, on the velocity $\mathbf{v}_i$ and the control input $\mathbf{a}_{d,i}$ of all vehicles:

$$0 \le \|\mathbf{v}_i^h(k)\| \le v_{max}^h \tag{3}$$

$$-v_{max}^z \le v_i^z(k) \le v_{max}^z \tag{4}$$

$$0 \le \|\mathbf{a}_{d,i}^h(k)\| \le a_{max}^h \tag{5}$$

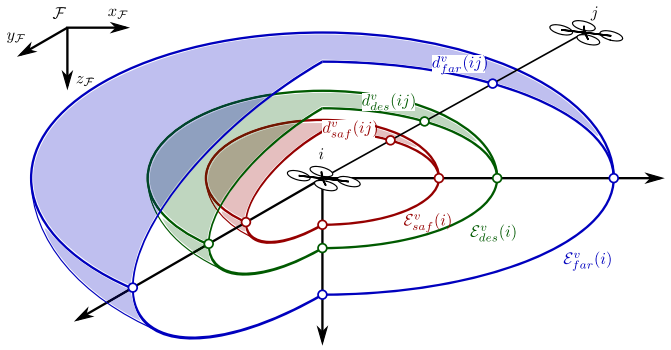

Fig. 1. Safety ($\mathcal{E}_{\text{saf}}^v(i)$), desired-locations ($\mathcal{E}_{\text{des}}^v(i)$), and remoteness ($\mathcal{E}_{\text{far}}^v(i)$) ellipsoids around vehicle i.

$$-a_{max}^z \le a_{d,i}^z(k) \le a_{max}^z \tag{6}$$

Our objective is to guide the vehicles to reach N^w way-points located at positions $\mathbf{p}_w$ ($w = 1, 2, \ldots, N^w$) in this predefined order. A way-point is reached as soon as its distance to one of the vehicles becomes lower than d_{vis}. After that, all vehicles consider the next way-point of the list. No feasible path has been previously determined to reach the way-points.

To succeed in their mission, vehicles must also avoid collisions with each other and external obstacles. At last, vehicles must, when possible, travel together and at nominal velocity v_n.

3. Control strategy

The apparition of cooperation is based on two features. The first is the communication of information between vehicles. The main piece of information shared is the predicted position $\hat{\mathbf{p}}_i$ of each vehicle, given in the common inertial frame $\mathcal{F}$. Any other piece of information acquired by a vehicle (*e.g.* position of external obstacle) is also transmitted to the rest of the group.

The second feature consists in considering these predicted positions when computing one's control input. This piece of information allows each vehicle to regulate its distance to the other vehicles to complete the objectives of collision avoidance and travelling together.

A safety ellipsoid, denoted $\mathcal{E}_{\text{saf}}^v(i)$, is defined around each vehicle i to ensure collision avoidance (see Fig. 1). If a vehicle enters this zone, it is considered as a collision with vehicle i. A safety zone denoted $\mathcal{E}_{\text{saf}}^o(o)$ is also defined around each obstacle o. The vehicles must avoid entering these as well.

To take an additional safety margin for obstacle avoidance, a desired obstacle passing distance greater than the safety zone is defined by $\mathcal{E}_{\text{des}}^o(o)$.

To travel as a group, each vehicle should be located at a desired distance from the others. As this desired distance can be different for horizontal and vertical components, an ellipsoid is used to represent desired locations around vehicles. This ellipsoid around vehicle i is denoted $\mathcal{E}_{\text{des}}^v(i)$ (see Fig. 1).

When a vehicle gets separated of the others by a large distance, it is not considered to be part of the group any more. This border is materialised by an ellipsoid $\mathcal{E}_{\text{far}}^v(i)$ for each vehicle i. Vehicles that are outside this ellipsoid are ignored by i. This represents the loss of communication and sensing that arises when vehicles are too distant. A representation of these ellipsoids is given in Fig. 1.

To preserve scalability and ensure robustness to single vehicle failure, each vehicle must compute its own control input within the time of an iteration. This computation is done using an MPC approach because it allows one to easily take the predicted positions of the vehicles into account. Moreover, the objective function used in the MPC approach gives the possibility to arbitrate between the several objectives of the mission.

4. Model Predictive Control (MPC)

Model Predictive Control (MPC) consists in using a kinematic model of the system to be controlled to predict over a prediction horizon of length H_p the effect of a given control sequence of length $H_c \leq H_p$. This prediction ability is used to find, using a numerical optimisation procedure at regular time intervals, the control sequence that minimises a performance criterion while respecting constraints on the state and the control input of the system.

The first element of the computed control sequence is then applied as control input of the system. The state of the system, on which predictions are based, is also updated and will be taken into account at the next iteration.

In our study, in order to comply to the desired distributed scheme, each vehicle i executes, at each time step k and synchronously with other vehicles, its own optimisation procedure to find its own control input. Control problem $\mathcal{P}_i(k)$ solved simultaneously by each vehicle i at time step k is described as:

$\mathcal{P}_i(k)$: *find* $\mathbf{a}^*_{d,ik}{}^{k+H_c-1} = \{\mathbf{a}^*_{d,i}(k), \mathbf{a}^*_{d,i}(k+1), \ldots, \mathbf{a}^*_{d,i}(k+H_c-1)\}$, *the control sequence of length* H_c *that minimises the following cost function, evaluated over the prediction horizon* H_p:

$$J_i^{rhc}(k, \mathbf{x}_i(k), \widehat{\mathbf{p}_{-i}}(_{k+1}^{k+H_p-1}) \big| k-1), \widehat{\mathbf{a}_{d,i}}(_{k}^{k+H_c-1}) \big| k))$$

under the constraints (3)–(6).

$\widehat{\mathbf{p}_{-i}}(_{k+1}^{k+H_p-1}) \big| k-1)$ denotes the predicted (symbolised by $\widehat{\bullet}$) positions of all vehicles except i (symbolised by $\bullet_{-i}$) between instants $k+1$ and $k+H_p-1$. These predictions are obtained from the optimisation procedures of other vehicles at time step $k-1$ (symbolised by $\widehat{\bullet}(\bullet|k-1)$). At each time step, after their computation, these predictions are shared by vehicles and are thus known after a delay of one iteration. Hence they are available up to time step $k+H_p-1$ only.

The predicted state $\widehat{\mathbf{x}}_i(_{k+1}^{k+H_p}|k)$ of vehicle i from time step $k+1$ to $k+H_p$ is computed from the current state $\mathbf{x}_i(k)$ at step k and the evaluated control sequence $\widehat{\mathbf{a}_{d,i}}(_{k}^{k+H_c-1}|k)$ using the model of the vehicles given by Eqs. (1) and (2). From $k+H_c$ to $k+H_p$, state of vehicle i is predicted assuming that the applied control is null ($\widehat{\mathbf{a}_{d,i}}(k+H_c|k) = \widehat{\mathbf{a}_{d,i}}(k+H_c+1|k) = \cdots = \widehat{\mathbf{a}_{d,i}}(k+H_p-1|k) = 0$).

Assuming that this control problem can be solved in due time, the MPC approach consists of the following three steps, repeated at each time step until the mission is accomplished.

1. each vehicle i computes its control sequence $\mathbf{a}^*_{d,i}(_{k}^{k+H_c-1}|k)$,
2. each vehicle then applies the first element of its control sequence, $\mathbf{a}^*_{d,i}(k|k)$,
3. each vehicle communicates the corresponding predicted positions to the other vehicles.

5. Objective function definition

The performance of an MPC strategy depends largely on the choice of its objective function $J_i^{rhc}(k, \mathbf{x}_i(k), \widehat{\mathbf{p}_{-i}}(_{k+1}^{k+H_p-1}) \big| k-1), \widehat{\mathbf{a}_{d,i}}(_{k}^{k+H_c-1}) \big| k))$.

To consider the several aspects of the mission and balance between them, the objective function can be defined as a weighted sum of several subcosts or components. Each component represents the completion of one aspect of the mission.

These components are divided into four main categories, as proposed in (7). The first is the control cost J_i^u that aims at minimising the amplitude of the control inputs and thus the energy consumption. The second is the manoeuvre cost J_i^{ma}, its purpose is to regulate the manoeuvres of the vehicles. The third is the mission cost J_i^{mis}, its role is to promote mission completion. The last category is the safety cost J_i^{saf} that ensures collision avoidance:

$$J_i^{rhc}(k) = J_i^u(k) + J_i^{ma}(k) + J_i^{mis}(k) + J_i^{saf}(k) \tag{7}$$

5.1. Control cost J_i^u

This cost is defined in a usual quadratic form by (8), the weighting coefficients $W^\bullet$ are defined in Section 5.5:

$$J_i^u(k) = \sum_{n=k+1}^{k+H_c} \widehat{\mathbf{a}_{d,i}}(n|k)^t . \begin{bmatrix} W^{u,h} & 0 & 0 \\ 0 & W^{u,h} & 0 \\ 0 & 0 & W^{u,z} \end{bmatrix} . \widehat{\mathbf{a}_{d,i}}(n|k) \tag{8}$$

Note that the choice of identical weights $W^{u,h}$ for the two horizontal directions is consistent with constraint (5) and does not favour any special direction in the horizontal plane.

5.2. Manoeuvre cost J_i^{ma}

As the considered control inputs consist of the accelerations of the vehicle along the three axes, the control cost does not take the current velocity and position of the vehicle into account. The manoeuvre cost purpose is to regulate vehicles' trajectories.

This cost, J_i^{ma}, is composed of three elements, as defined in (9). $J_i^{ma,norm}$, defined by (10), urges the vehicle to move at nominal velocity v_n on the horizontal plan whereas $J_i^{ma,alti}$, defined by (11), tends to maintain the vehicle at its current altitude. $J_i^{ma,rot}$ finally, defined by (12), penalises changes of velocity direction.

Let us recall that $\mathbf{v}_i^h$ and $\mathbf{a}_{d,i}^h$ denote respectively the projection of vehicle i velocity vector and control input on the horizontal plane and that v_i^z is the vertical velocity of vehicle i. The weighting coefficients $W^\bullet$ are defined in Section 5.5:

$$J_i^{ma}(k) = J_i^{ma,norm}(k) + J_i^{ma,alti}(k) + J_i^{ma,rot}(k) \tag{9}$$

$$J_i^{ma,norm}(k) = W^{ma,norm} . \sum_{n=k+1}^{k+H_c} (\|\widehat{\mathbf{v}_i^h}(n|k)\| - v_n)^2 \tag{10}$$

$$J_i^{ma,alti}(k) = W^{ma,alti} . \sum_{n=k+1}^{k+H_c} \|\widehat{v_i^z}(n|k)\|^2 \tag{11}$$

$$J_i^{ma,rot}(k) = \begin{cases} 0 & \text{if } \|\mathbf{v}_i^h(k)\| = 0 \\ W^{ma,rot} . \dfrac{\|\mathbf{v}_i^h(k) \times \widehat{\mathbf{a}_{d,i}^h}(k|k)\|^2}{\|\mathbf{v}_i^h(k)\|^2} & \text{if } \mathbf{v}_i^h(k) . \widehat{\mathbf{a}_{d,i}^h}(k|k) \geq 0 \\ W^{ma,rot} . \left(2 . \|\widehat{\mathbf{a}_{d,i}^h}(k|k)\|^2 - \dfrac{\|\mathbf{v}_i^h(k) \times \widehat{\mathbf{a}_{d,i}^h}(k|k)\|^2}{\|\mathbf{v}_i^h(k)\|^2} \right) & \text{if } \mathbf{v}_i^h(k) . \widehat{\mathbf{a}_{d,i}^h}(k|k) < 0 \end{cases} \tag{12}$$

This objective function will be denoted $J_i^{rhc}(k)$ in the rest of the paper.

Remark on $J_i^{ma,rot}$: The basic idea is to minimise the component of $\widehat{\mathbf{a}_{d,i}^h}$ that is perpendicular to $\mathbf{v}_i^h$ as this is what causes the vehicle

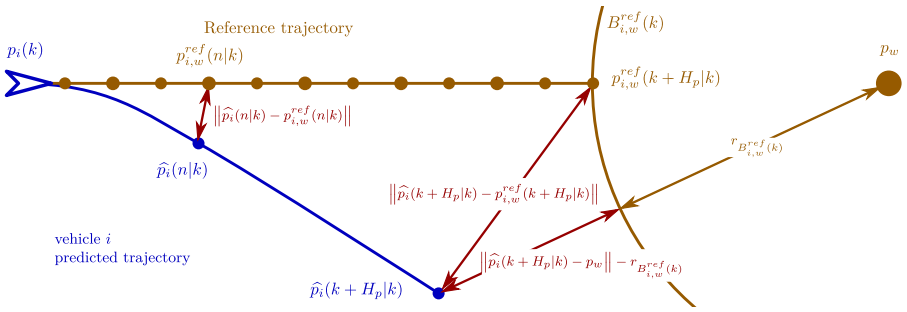

Fig. 2. Representation of reference points $\mathbf{p}_{i,w}^{ref}$ and reference ball $B_{i,w}^{ref}$.

to turn. The case where $\mathbf{v}_i^h \cdot \widehat{\mathbf{a}_{d,i}^h} < 0$ is separated so this element does not favour slowing down over turning.

5.3. Mission costs J_i^{mis}

This cost is composed of three elements as defined in (13). The first, $J_i^{mi,direct}$ makes vehicles move along straight line reference trajectories toward the next way-point. The second, $J_i^{mi,final}$ drives vehicles closer to the way-point. The third, $J_i^{mi,flock}$ makes vehicles group together:

$$J_i^{mis}(k) = J_i^{mi,direct}(k) + J_i^{mi,final}(k) + J_i^{mi,flock}(k) \tag{13}$$

Each element is now detailed:

Definition of $J_i^{mi,direct}$: The purpose of this cost is to make vehicle i move in a straight line toward the way-point to be reached w. To do that, reference points are build between the vehicle and this way-point to define a straight line reference trajectory as illustrated in Fig. 2. Each reference point $\mathbf{p}_{i,w}^{ref}(n|k)(n \in [k+1, k+H_p])$ is placed at the location that the vehicle i would reach at step n, if it were to move toward the way-point w in a straight line, at the nominal velocity v_n, and regardless of its constraints. The positions of the reference points are given by (14) and $J_i^{mi,direct}$ is defined by (15):

$$\mathbf{p}_{i,w}^{ref}(n|k) = \mathbf{p}_i(k) + (n-k)\cdot\Delta t \cdot v_n \cdot \frac{\mathbf{p}_i(k) - \mathbf{p}_w}{\|\mathbf{p}_i(k) - \mathbf{p}_w\|} \quad \forall n \in [k+1, k+H_p] \tag{14}$$

$$J_i^{mi,direct}(k) = W^{mi,direct} \cdot \sum_{n=k+1}^{k+H_p} \|\widehat{\mathbf{p}}_i(n|k) - \mathbf{p}_{i,w}^{ref}(n|k)\|^2 \tag{15}$$

The weighting coefficient $W^{mi,direct}$ is defined in Section 5.5.

Definition of $J_i^{mi,final}$: The purpose of this cost is to move vehicle i closer to the way-point w by the end of the prediction horizon H_p. To do that, a reference ball $B_{i,w}^{ref}(k)$, illustrated in Fig. 2, is defined as the smallest ball around w that vehicle i can hypothetically reach from its current position by moving directly toward this way-point at nominal velocity v_n. The radius of the reference ball is given by (16), and $J_i^{mi,final}$ is defined by (17), where $d_{iw}(k) = \|\mathbf{p}_w - \mathbf{p}_i(k)\|$ denotes the current distance between the vehicle i and the way-point w and $d(\widehat{\mathbf{p}}_i(k+H_p|k), B_{i,w}^{ref}(k))$ denotes the distance between the predicted position $\widehat{\mathbf{p}}_i(k+H_p|k)$ of vehicle i at step $k+H_p$ and the ball $B_{i,w}^{ref}(k) = \{\mathbf{x} | \|\mathbf{x} - \mathbf{p}_w\| \le r(B_{i,w}^{ref}(k))\}$ where $r(B_{i,w}^{ref}(k))$ is defined as

$$r(B_{i,w}^{ref}(k)) = \begin{cases} 0 & \text{if } d_{iw}(k) \le H_p \cdot \Delta t \cdot v_n \\ d_{iw}(k) - H_p \cdot \Delta t \cdot v_n & \text{otherwise} \end{cases} \tag{16}$$

$$J_i^{mi,final}(k) = W^{mi,final} \cdot (\|\widehat{\mathbf{p}}_i(k+H_p|k) - \mathbf{p}_w\| - r(B_{i,w}^{ref}(k)))^2 \tag{17}$$

The weighting coefficient $W^{mi,final}$ is defined in Section 5.5.

Remark 1. The expressions of $J_i^{mi,direct}$ and $J_i^{mi,final}$ are chosen so that these costs have a constant variation range. This allows one to keep the relative importance of the various costs constant throughout the mission, whatever the distance to the current way-point.

Remark 2. $J_i^{mi,direct}$ and $J_i^{mi,final}$ are complementary. Indeed, $J_i^{mi,direct}$ incites the vehicles to follow a straight line toward the next way-point instead of turning around while approaching (*i.e.* spiralling). But if an obstacle is located between a vehicle and the way-point, $J_i^{mi,direct}$ incites the vehicle to stop in front of the obstacle to deviate as little as possible from the reference trajectory. $J_i^{mi,final}$ on the other hand incites the vehicle to go around the obstacle to get closer to the way-point.

Definition of $J_i^{mi,flock}$: The purpose of this cost is to form a flock of vehicles. To do that, it penalises the predicted distance $\widehat{d_{ij}}(n|k) = \|\widehat{\mathbf{p}}_j(n|k) - \widehat{\mathbf{p}}_i(n|k)\|$ between vehicles i and j $(i \neq j)$. $J_i^{mi,flock}$ is defined by

$$J_i^{mi,flock}(k) = W^{mi,flock} \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \sum_{n=k+1}^{k+H_p} \frac{1 + \tanh(\widehat{d_{ij}}(n|k) - \beta_{ij}^f) \cdot \alpha_{ij}^f)}{2} \tag{18}$$

The weighting coefficient $W^{mi,flock}$ is defined in Section 5.5, α^f and β^f are defined below.

The choice to base the form of $J_i^{mi,flock}$ on a hyperbolic tangent stems from several constraints. Firstly, two vehicles i and j must not be encouraged to be closer than the distance defined by the ellipsoid $\mathcal{E}_{des}^v(i)$, therefore the cost must vary a little under this threshold. Secondly, it has been stated in Section 3 that vehicles farther than the ellipsoid $\mathcal{E}_{far}^v(i)$ must be ignored by vehicle i. Accordingly, the cost must vary little over this threshold. Finally, between these two areas, it is desirable that the cost varies smoothly. The hyperbolic tangent satisfies these properties.

The coefficients α_{ij}^f and β_{ij}^f are used to shape the hyperbolic tangent according to our needs: α_{ij}^f defines the width of the region where the function varies rapidly and β_{ij}^f defines the position of this region in the abscissa axis. These terms depend on the relative position of the vehicles i and j. For given positions of vehicles i and j, the ellipsoids $\mathcal{E}_{des}^v(i)$ and $\mathcal{E}_{far}^v(i)$ respectively define the distances $d_{des}^v(ij)$ and $d_{far}^v(ij)$ as seen in Fig. 1. The terms α_{ij}^f and β_{ij}^f are derived from these constraints and given by

$$\beta_{ij}^f = 6.(d_{far}^v(ij) - d_{des}^v(ij))^{-1} \tag{19}$$

$$\alpha_{ij}^f = (d_{far}^v(ij) + d_{des}^v(ij))/2 \tag{20}$$

The definitions of α_{ij}^f and β_{ij}^f have been chosen in order to obtain a nearly constant cost inside $\mathcal{E}_{des}^v(i)$ and outside $\mathcal{E}_{far}^v(i)$ (defined by a derivative inferior to 0.05) and a symmetric behaviour at borders. Fig. 3 depicts this cost as a function of the distance between the vehicle i and the vehicle j.

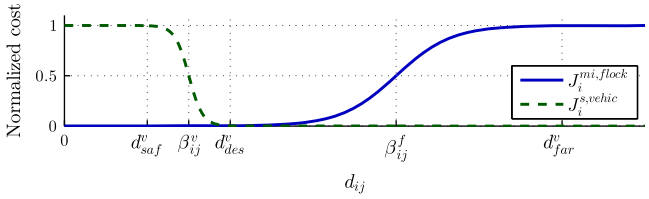

Fig. 3. Flocking ($J_i^{mi,flock}$) and avoidance ($J_i^{saf,vehic}$) costs.

5.4. Safety cost J_i^{saf}

This cost is composed of three elements as defined in (21). The first, $J_i^{saf,vehic}$, defined by (22), aims at avoiding collisions with other vehicles. The second, $J_i^{saf,obstac}$, defined by (23), aims at avoiding obstacles. The third, $J_i^{saf,trajec}$, defined by (24), penalises the difference between the trajectory that was transmitted to the other vehicles at the previous iteration $\widehat{\mathbf{p}}_i(_{k+1}^{k+H_p-1}|k-1)$ and the new predicted trajectory $\widehat{\mathbf{p}}_i(_{k+1}^{k+H_p-1}|k-1)$:

$$J_i^{saf}(k) = J_i^{saf,vehic}(k) + J_i^{saf,obstac}(k) + J_i^{saf,trajec}(k) \tag{21}$$

$$J_i^{saf,vehic}(k) = W^{saf,vehic} \cdot \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{n=k+1}^{k+H_p} \frac{1-\tanh((\widehat{d_{ij}}(n|k) - \beta_{ij}^v).\alpha_{ij}^v)}{2} \tag{22}$$

$$J_i^{saf,obstac}(k) = W^{saf,obstac} \cdot \sum_{o=1}^{N^o} \sum_{n=k+1}^{k+H_p} \frac{1-\tanh((\widehat{d_{io}}(n|k) - \beta_{io}^o).\alpha_{io}^o)}{2} \tag{23}$$

$$J_i^{saf,trajec}(k) = W^{saf,trajec} \cdot \sum_{n=k+1}^{k+H_p-1} \|\widehat{\mathbf{p}}_i(n|k) - \widehat{\mathbf{p}}_i(n|k-1)\|^2 \tag{24}$$

The weighting coefficients $W^\bullet$ are defined in Section 5.5, N^o is the number of obstacles, $\widehat{d_{io}}(n|k)$ is the distance separating the predicted position $\widehat{\mathbf{p}}_i(n|k)$ of vehicle i from the obstacle o at step n. The hyperbolic tangent is chosen for the same reasons as stated before. The terms $\alpha^\bullet$ and $\beta^\bullet$ are given by (25)–(28) similar to (19) and (20). The values of desired and safety distances (respectively $d^v_{\text{des}}(ij)$ and $d^v_{\text{saf}}(io)$ between vehicles, $d^o_{\text{des}}(ij)$ and $d^o_{\text{saf}}(io)$ with obstacles) are defined using the corresponding ellipsoids as illustrated in Fig. 1:

$$\beta_{ij}^v = 6.(d^v_{\text{des}}(ij) - d^v_{\text{saf}}(ij))^{-1} \tag{25}$$

$$\alpha_{ij}^v = (d^v_{\text{des}}(ij) + d^v_{\text{saf}}(ij))/2 \tag{26}$$

$$\beta_{io}^o = 6 \cdot (d^o_{\text{des}}(io) - d^o_{\text{saf}}(io))^{-1} \tag{27}$$

$$\alpha_{io}^o = (d^o_{\text{des}}(io) + d^o_{\text{saf}}(io))/2 \tag{28}$$

5.5. Definition of weighting coefficients $W^\bullet$

Each component of the objective function is weighted according to its relative priority, that is the importance of the corresponding task in the mission. As an example, remaining group could be more important than travelling at nominal velocity but less important than collision avoidance. Therefore, the group may be split to avoid collision, but otherwise vehicles would adapt their velocity to remain together.

The weights $W^\bullet = w^\bullet \cdot k^\bullet$ consist of:

- $k^\bullet$: a normalisation coefficient, used to ensure that all components of the objective function have the same order of magnitude;
- $w^\bullet$: the tuning parameter used to control the relative importance of the normalised components of the objective function.

Table 1
Forms of normalisation coefficients.

Coefficient	Form
$k^{u,h}$	$(H_c \cdot (a^h_{max})^2)^{-1}$
$k^{u,z}$	$(H_c \cdot (a^z_{max})^2)^{-1}$
$k^{ma,norm}$	$(H_c \cdot (v^h_{max} - v_n)^2)^{-1}$
$k^{ma,alti}$	$(H_c \cdot (v^z_{max})^2)^{-1}$
$k^{ma,rot}$	$((a^h_{max})^2)^{-1}$
$k^{mi,direct}$	$(\sum_{n=1}^{H_p}(n \cdot \Delta t \cdot v_n)^2)^{-1}$
$k^{mi,final}$	$((H_p \cdot \Delta t \cdot v_n)^2)^{-1}$
$k^{mi,flock}$	$(H_p \cdot N)^{-1}$
$k^{saf,vehic}$	$(H_p/2)^{-1}$
$k^{saf,obstac}$	$(H_p/2)^{-1}$
$k^{saf,trajec}$	$(\sum_{n=1}^{H_p}(n \cdot \Delta t \cdot v_n)^2)^{-1}$

To define the normalisation coefficients, reference scenarios are used whose normalised cost must be one. For $J_i^{mi,direct}$, $J_i^{mi,final}$ (progression toward the way-point) and $J_i^{saf,trajec}$ (small deviations), a scenario where the vehicle does not move is used. For $J_i^{mi,flock}$ (travelling as a group) a scenario where the vehicle ignores all the others is used. For the remaining of the coefficients, this scenario is defined as the worst movement of vehicle i that does not compromise the mission (*e.g.* highest acceleration allowed for the entire control horizon for $k^{u,h}$).

Normalisation coefficients are given in Table 1.

6. Control sequence search procedure

In this approach, a search procedure is used instead of traditional optimisation to solve the control problem $\mathcal{P}_i(k)$. Instead of running an optimisation algorithm until a minimum of the cost function is found, which would be computationally expensive, the chosen search procedure consists in finding the control sequence that yields the minimum cost among a predefined set of candidates $\mathcal{S}$.

This strategy does not provide the optimal control sequence but, if the elements of $\mathcal{S}$ are chosen wisely, it can provide a near optimal feasible control sequence. This strategy also has two advantages over a traditional optimisation procedure. Firstly, the amount of computation necessary to find a control sequence is constant in all situations, hence the computation delay is constant. This is of particular interest when faced to a collision risk, when decision has to be taken fast. The second advantage is that the strategy of systematic search is less sensitive to problems of local minima as the entire control space is explored. Finally, the systematic search requires no initialisation.

6.1. Search procedure

The studied search procedure consists in defining, prior to the mission, a set of candidate control sequences. This definition will be explained in Section 6.3. At each time step, the control problem $\mathcal{P}_i(k)$ is solved using the proposed search procedure as follows:

1. using a model of the vehicles dynamics, predict the effect of each control sequence of the set of candidates $\mathcal{S}$ on the state of the vehicle;
2. remove from $\mathcal{S}$ all the candidate control sequences that lead to violation of constraints on the state of the vehicule (3) and (4);
3. compute the cost $J_i^{rhc}(k)$ corresponding to each remaining candidate control sequence;
4. select the control sequence that implies the smallest cost.

Remark 3. As explained is Section 6.3, the candidate control sequences of the set $\mathcal{S}$ satisfy, by constructions the control constraints of the vehicles (5) and (6). Additionally, thanks to step 2, each candidate of the set $\mathcal{S}$ respects the velocity constraints of the vehicles (3) and (4). Therefore, each control sequence of $\mathcal{S}$ is guaranteed to be feasible, and thus, the selected control sequence is also feasible. As the weights on the collision cost is higher than the others, the selection of the control input is indeed prone to favour a control sequence avoiding collision. However, as the cost consists of several sub-costs, it is not absolutely guaranteed that the selected sequence will avoid any collision. In the set $\mathcal{S}$, the selection of the control inputs is only performed so as to rule out control sequences which could not be executed by a vehicle, according to constraints (3)–(6).

Remark 4. The time needed to find the smallest value among the discrete set of costs corresponding to each element of $\mathcal{S}$ is negligible compared to the time necessary to predict the trajectories and compute the costs.

6.2. Reducing the amount of computation needed

Because all the candidate control sequences of the set $\mathcal{S}$ are evaluated, it is necessary to find ways to limit the amount of computation needed. Three means are used to that end.

Firstly, instead of using an actual, accurate model of the vehicles, the model used to predict the trajectories of the vehicles is the simple model defined in Section 2 by Eqs. (1) and (2).

A second means to reduce the amount of computation needed is to keep the control input of the vehicles constant during the entire control horizon H_c. The control is then supposed null for the remainder of the prediction horizon H_p, which means that the vehicle velocity is constant. With this method, the number of control inputs to be found goes from $3 \cdot H_c$ (acceleration along the three axes over the control horizon) to 3.

When small sampling time is used and control horizon is short, the effect on the predicted positions of a sequence of different controls can be approached very closely by using a constant control. This effect is due to the small distances covered by the vehicles during an iteration. This means that vehicle capabilities are not impaired by this simplification.

A third and last means to limit the amount of computation is to distribute the candidate control sequences of the set $\mathcal{S}$ in a particular way. The objective is to have the smallest possible number of elements, yet to explore the control space efficiently. This aspect is explained in the next section.

6.3. Distribution of the control sequences

As stated in Section 6.2, the control sequences consist of a constant control input during the control horizon, followed by a null control input for the remaining of the prediction horizon.

The efficiency in our method relies on the distribution of these control inputs over the control space. As our control inputs are the accelerations along the three axes of the vehicles, the control space is three dimensional. At first glance, three possible ways to distribute the control inputs are:

1. generate many control sequences distributed uniformly over the control space. This allows precise control, but takes time to predict the effect of all the control sequences;
2. generate a moderate amount of uniformly distributed control sequences. This will be fast but the control may lack precision and cause oscillations or missed possibilities;
3. generate a moderate amount of control sequences randomly distributed over the control space. In this approach, the set is redefined at each time step. This is the approach of Frew (2006) (but with constant velocity and a single vehicle). It is fast and as the distribution changes iteration after iteration, the probability to find the control that will allow the discovery of a new, better trajectory increases. A set of predefined control sequences must be added to the randomly generated set to ensure that particular trajectories are always possible (like straight line, or maximum turn rate).

In this work, another approach is chosen which uses a small amount of control sequences distributed in a particular way over the control space. The chosen distribution intends to implement the following intuition:

At the beginning of a high amplitude manoeuvre (like a u-turn or an emergency brake), a precise control input is not necessary because the amplitude of control is the main concern. On the other hand, as manoeuvre comes to an end, or for small amplitude manoeuvre, higher precision is desirable.

This intuition motivates the three following rules.

1. the set $\mathcal{S}$ of candidates includes the extreme control inputs ($\pm a^h_{max}$ and $\pm a^z_{max}$) to exploit the full potential of the vehicles;
2. the set $\mathcal{S}$ of candidates includes the null control input (that is $a^x_d = 0$, $a^y_d = 0$, and $a^z_d = 0$) to allow one to continue with the same velocity;
3. candidates are distributed over the entire control space with an increased density around the null control input.

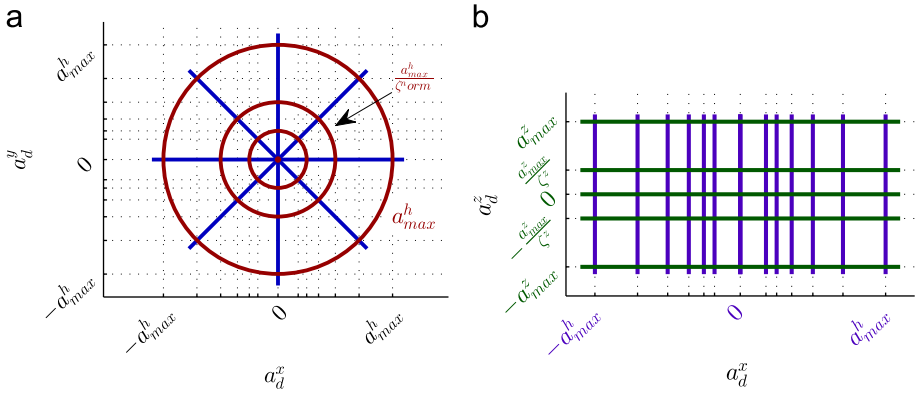

Fig. 4. Illustration of the set $\mathcal{S}$ of candidate control sequences ($\zeta^{norm} = 2$, $\zeta^z = 3$, $N^{dir} = 8$, $N^{norm} = 3$ and $N^z = 5$). Projection on the (a) $(x_\mathcal{F}, y_\mathcal{F})$ plane. (b) $(x_\mathcal{F}, z_\mathcal{F})$ plane.

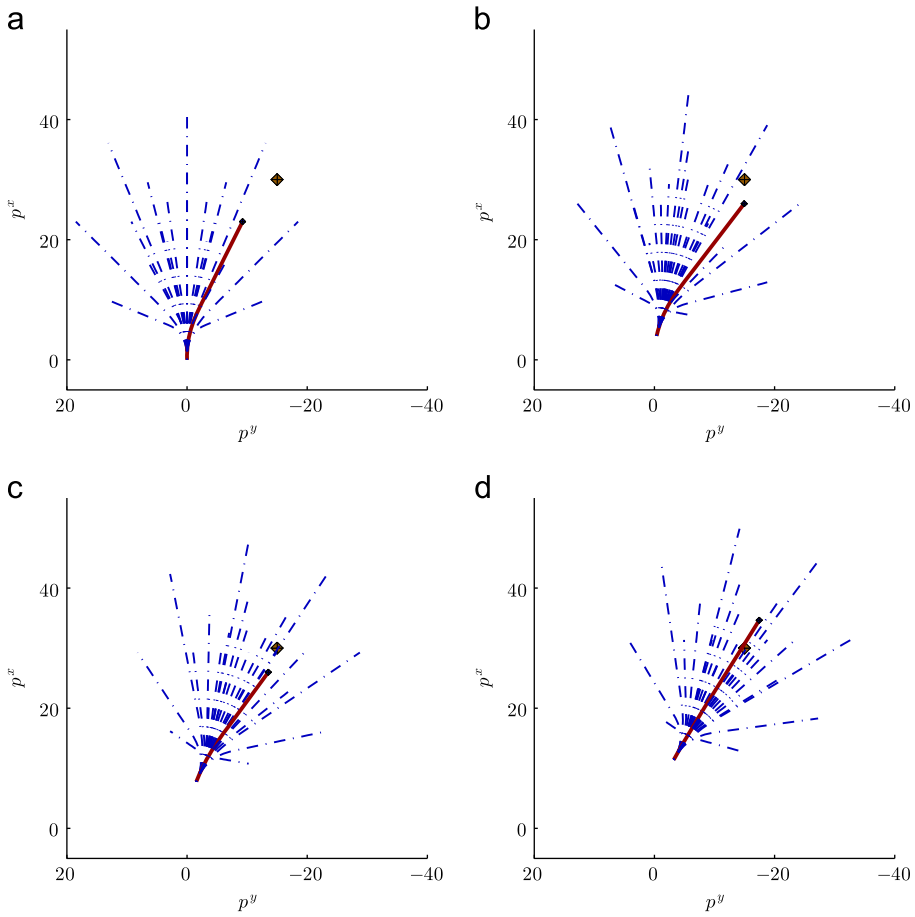

Fig. 5. Top view of the evolution of a vehicle aiming for a way-point (represented by a diamond), with its movements restricted to the candidate control inputs. The initial velocity of the vehicle is 2 m s^{-1} in the $x_{\mathcal{F}}$ direction. (a) Initial step. (b) Step 5. (c) Step 9. (d) Step 13.

Let us recall that the constraints on the control inputs are a maximal norm for its projection in the horizontal plane (5) and maximal and minimal values for its vertical component (6). To reflect these constraints, it has been chosen to define independently the sets of horizontal $\mathcal{S}^h$ and vertical $\mathcal{S}^z$ accelerations. Additionally, $\mathcal{S}^h$ is defined using a set of directions $\mathcal{S}^{dir}$ and a set of norms $\mathcal{S}^{norm}$, and subsequently converted to obtain the components along $x_{\mathcal{F}}$ and $y_{\mathcal{F}}$ of the acceleration. These sets are defined by

$$\mathcal{S} = \{\{\mathcal{S}^{dir} \times \mathcal{S}^{norm}\} \cup (0,0)\} \times \mathcal{S}^z \tag{29}$$

$$\mathcal{S}^{dir} = \left\{\frac{2.\pi.p}{\eta^{dir}}\right\} \quad \text{with } p = 1 \text{ to } \eta^{dir} \tag{30}$$

$$\mathcal{S}^{norm} = \left\{\frac{a^h_{max}}{(\zeta^{norm})^p}\right\} \quad \text{with } p = 0 \text{ to } \eta^{norm} \tag{31}$$

$$\mathcal{S}^z = \left\{\pm\frac{a^z_{max}}{(\zeta^z)^p}\right\} \cup \{0\} \quad \text{with } p = 0 \text{ to } \eta^z \tag{32}$$

ζ^{norm} and ζ^z control the interval between two candidates; the number of candidates N^c, N^{dir}, N^{norm}, and N^z in respectively $\mathcal{S}$, $\mathcal{S}^{dir}$, $\mathcal{S}^{norm}$, and $\mathcal{S}^z$ are deduced from η^{dir}, η^{norm}, and η^z using (33)–(36). The resulting complete set $\mathcal{S}$ is illustrated in Fig. 4a and b:

$$N^c = (N^{dir} \cdot N^{norm} + 1) \cdot N^z \tag{33}$$

$$N^{dir} = \eta^{dir} \tag{34}$$

$$N^{norm} = \eta^{norm} + 1 \tag{35}$$

$$N^z = 2 \cdot \eta^z + 1 \tag{36}$$

Fig. 5a–d shows how, with this choice of control inputs, a vehicle can aim at an arbitrary point. 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 $\zeta^\bullet$ and $\eta^\bullet$ parameters.

6.4. Discussion on the number of candidates

The particular distribution that has been chosen aims at exploring the control space efficiently, *i.e.* exploring all of it but insisting on the most useful part to reduce the amount of computation.

To explore systematically the control space, each element of $\mathcal{S}$ will be used to predict a trajectory and evaluate the associated cost. The amount of computation is thus directly proportional to the number of candidates $N^c = (N^{dir} \cdot N^{norm} + 1) \cdot N^z$.

The values of N^{dir}, N^{norm} and N^z must be chosen while considering different points:

- precision of the actuators: it is unnecessary to test two control inputs that will be executed in the same way;
- precision of the available measurements: it is unnecessary to test two control inputs whose executions will be undistinguishable;
- importance of precise guidance and computation capacity of the on-board computer: more precision means more computation delay.

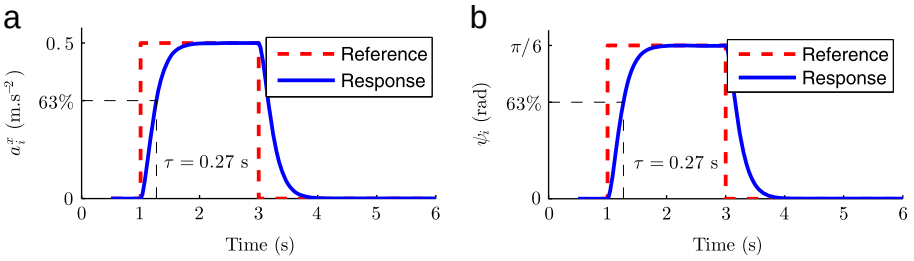

Fig. 6. Controlled quadrotor step response. (a) Step response to a desired acceleration along $x_{\mathcal{F}}$ of 0.5 m s^{-2}. (b) Step response to a desired yaw angle of $\pi/6$ rad. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this paper.)

7. Application to a flock of quadrotors

7.1. Quadrotor vehicles

Evaluation of the performances of the proposed approach in a realistic context is performed, in simulation, using a flock of miniature four-rotor helicopters derived from the description given in Mokhtari and Benallegue (2004). The simulated dynamics of the vehicles is derived from the application of Newton laws to a realistic model of vehicle.

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 the opposite directions. The control input of the rotors is the signals u_i^1 to u_i^4 classically defined as:

- u_i^1: the resulting thrust of the four rotors (controls the movement along the z-axis of the vehicle);
- u_i^2: the difference of thrust between left and right rotors (controls the roll φ_i and hence contribute to the movement along the y-axis of the vehicle);
- u_i^3: the difference of thrust between front and back motors (controls the pitch θ_i and hence contribute to the movement along the x-axis of the vehicle);
- u_i^4: the difference of torque between the clockwise and anti-clockwise rotating rotors (controls the yaw ψ_i of the vehicle).

Let us denote by $\boldsymbol{\eta}_i(k) = [\psi_i(k)\ \theta_i(k)\ \varphi_i(k)]^t$ the attitude vector of vehicle i and by $\mathbf{x}_i(k) = [\mathbf{p}_i(k)^t\ \mathbf{v}_i(k)^t\ \boldsymbol{\eta}_i(k)^t\ \dot{\boldsymbol{\eta}}_i(k)^t]^t$ its state vector.

It is assumed that at each iteration k the value of the state $\mathbf{x}_i(k)$ is available for computation of vehicle i control. The control strategy consists in applying the MPC guidance law based on the simplified prediction model (1) and (2), described in Section 4, to compute a desired acceleration vector $\mathbf{a}_{d,i}(k) = [a_{d,i}^x(k) a_{d,i}^y(k) a_{d,i}^z(k)]^t$. This desired acceleration, along with a given desired value $\psi_{d,i}$ for the yaw, is then converted into vehicle control inputs $u_i^1(k)$ to $u_i^4(k)$. Attitude control of the quadrotor is then achieved by using the approach proposed in Mokhtari and Benallegue (2004).

Illustrations of the response of the controlled vehicle to a step of acceleration in the $x_{\mathcal{F}}$ direction and to a step of yaw angle are given in Fig. 6. The desired value is presented by the red dotted line whereas the simulated response is the plain blue line.

7.2. Mission simulation

A flock of $N=7$ quadrotors simulated by the model exposed in Section 7.1 and guided using the proposed MPC scheme with the prediction model (1) and (2) is now considered. In the simulated mission, the flock must successively reach three way-points while avoiding obstacles and collisions. The vehicles also have to travel as a group at nominal velocity v_n. Let us recall that the $z_{\mathcal{F}}$-axis is directed downward. As the coordinate of the ground is $z_{\mathcal{F}} = 0$, the altitude of a vehicle is given by $-z_{\mathcal{F}}$. At all times, the vehicles must fly between the altitudes of 0 m (the ground) and 25 m. These constraints are materialised with two obstacles of infinite dimensions, with a vertical safety distance of 2 m.

Vehicles are initially placed randomly in the volume defined by $x_{\mathcal{F}} \in [-205\text{ m}, -155\text{ m}]$, $y_{\mathcal{F}} \in [-45\text{ m}, 5\text{ m}]$, $z_{\mathcal{F}} \in [-15\text{ m}, -5\text{ m}]$. Their initial velocities, attitudes and attitude derivatives are set to zero, the desired yaw angle ψ_d is also set to zero for all vehicles throughout the entire mission.

The values of the constraints on velocity and acceleration are given in Table 2a–c, whereas the ellipsoids defining the

Table 2
Vehicles parameters.

(a) Velocity constraints	
v_{max}^h	5 m s^{-1}
v_{max}^z	1 m s^{-1}
(b) Acceleration constraints	
a_{max}^h	0.5 m s^{-2}
a_{max}^z	0.25 m s^{-2}
(c) Other parameters	
Δt	0.5
v_n	2 m s^{-1}

Table 3
Parameters of the ellipsoids defining the characteristic distances.

Semi-axis	Along		
	$x_{\mathcal{F}}$	$y_{\mathcal{F}}$	$z_{\mathcal{F}}$
(a) Between vehicles			
$\mathcal{E}_{saf}^v$	10 m	10 m	5 m
$\mathcal{E}_{des}^v$	20 m	20 m	10 m
$\mathcal{E}_{far}^v$	50 m	50 m	25 m
(b) To obstacles			
$\mathcal{E}_{saf}^o$	4 m	4 m	2 m
$\mathcal{E}_{des}^o$	8 m	8 m	4 m

Table 4
Parameters of the search procedure.

(a) Horizons	
H_c	4
H_p	24
(b) Size of sets of control sequences	
N^{dir}	8
N^{norm}	3
N^c	125
N^z	5
(c) Repartition	
ζ^{norm}	2
ζ^z	3

Table 5
Weighting coefficients of the objective function.

(a) Control	
$w^{u,h}$	2
$w^{u,z}$	2
(b) Manoeuvre	
$w^{ma,norm}$	10
$w^{ma,alti}$	2
$w^{ma,rot}$	5
(c) Mission	
$w^{mi,direct}$	10
$w^{mi,final}$	20
$w^{mi,flock}$	50
(d) Safety	
$w^{saf,vehic}$	100
$w^{saf,obstac}$	400
$w^{saf,trajec}$	0

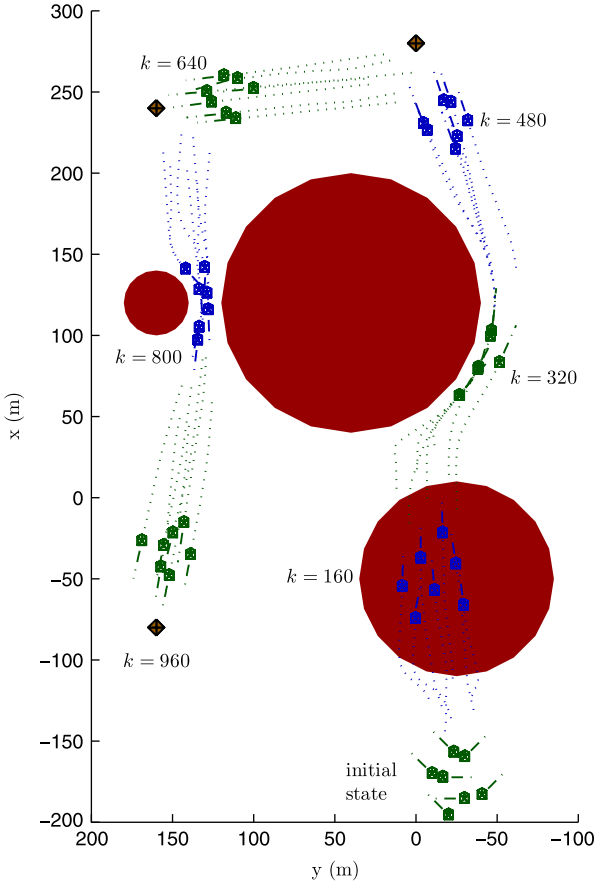

Fig. 7. Top view of the trajectories followed by the vehicles to complete their mission (way-points are represented by diamonds and obstacles by cylinders).

characteristic distances are defined in Table 3a and b. The parameters of the search procedure are given in Table 4a–c and finally, the tuning parameters of the objective function of the MPC approach are given in Table 5a–d. Let us recall that they define the relative importance of each component of the mission.

Fig. 7 presents the environment of the mission and the resulting trajectories of the vehicles.

7.3. Mission analysis

The mission presented in Fig. 7 is considered to be a success, because all way-points have been successfully reached and

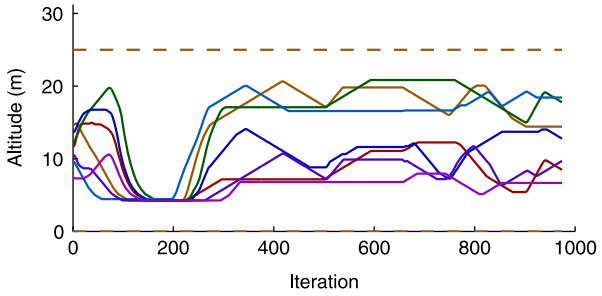

Fig. 8. Evolution of the altitudes of the vehicles (the dotted lines represent the altitude constraints introduced in Section 7.2).

vehicles have remained grouped together and avoided 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 $x_{\mathcal{F}}$-axis (*i.e.* $\psi_i(k) = 0$ for all i and k).

The computation delay has remained identical throughout the entire mission, even when vehicles faced obstacles.

Altitudes: The variation of the altitude of the vehicles throughout the mission is given Fig. 8. In this figure one can see that, excepted when they had to pass under the first obstacle, the vehicles spread over the vertical axis to form a tighter group while maintaining desired distances between them.

Distances: In Fig. 9, the distances between the vehicles and obstacles are compared to the safety, desired-locations, and remoteness ellipsoids. Fig. 9 shows that, during the mission, the vehicles have always managed to avoid entering the safety ellipsoid of obstacles or other vehicles. In Fig. 9a, it is also clear that the vehicles remained tightly grouped during the mission, except when they had to avoid obstacles.

Safety and mission costs: In Fig. 10, the evolution during the mission of several cost components is presented. The safety cost, depicted in the first two pictures, has increased each time vehicles had to avoid obstacle collisions. Thanks to this higher value, this cost overcame the other components and, as seen in Fig. 9, no collision occurs.

The mission costs $J^{mi,direct}$ and $J^{mi,final}$, that urge the vehicles to move toward the way-point, present several increases. While very short increases match the moments where the flock of vehicles reached a way point and switched to the next, longer increases in the mission cost correspond to the avoidance of obstacles. As previously stated in Remark 1, these two cost components play a complementary role. Indeed, while avoiding the second obstacle, $J^{mi,final}$ remains low, making the vehicles fly toward the way-point although $J^{mi,direct}$ becomes greater as the vehicles deviate from a straight line trajectory to the way-point.

The mission cost $J^{mi,flock}$, whose purpose is to group the vehicles together, presents increases at the same moments as safety cost. This is explained by the inclination of the vehicles to spread when confronted to obstacles in order to avoid collisions.

Constraints on the vehicles: In Figs. 11 and 12 it can be verified that the constraints on velocity and acceleration of the vehicles are satisfied throughout the mission.

7.4. Performance evaluation and comparison

Analysis proposed in this paragraph considers robustness to model mismatch, variations of initial conditions and selection of tuning parameters. It also compares the results with those obtained using other optimisation tools.

Model mismatch: For each simulation, the simulated vehicle model differs from the linear nominal model used for predictions

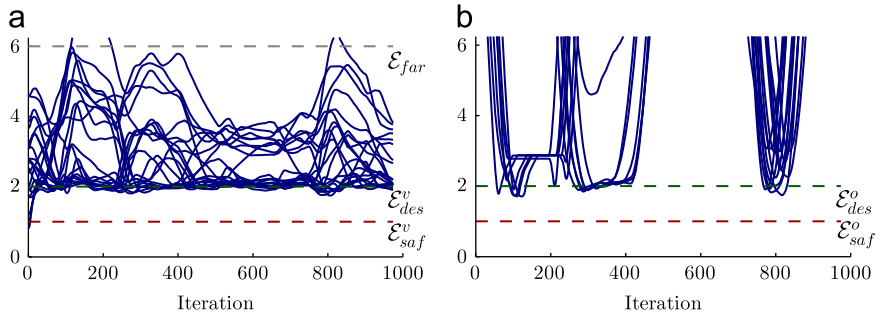

Fig. 9. Comparison of distances with the safety ($\mathcal{E}^v_{saf}, \mathcal{E}^o_{saf}$), desired-locations ($\mathcal{E}^v_{des}, \mathcal{E}^o_{des}$), and remoteness ($\mathcal{E}^v_{far}$) ellipsoids. (a) Distance between vehicles. (b) Distances from each vehicle to each obstacle.

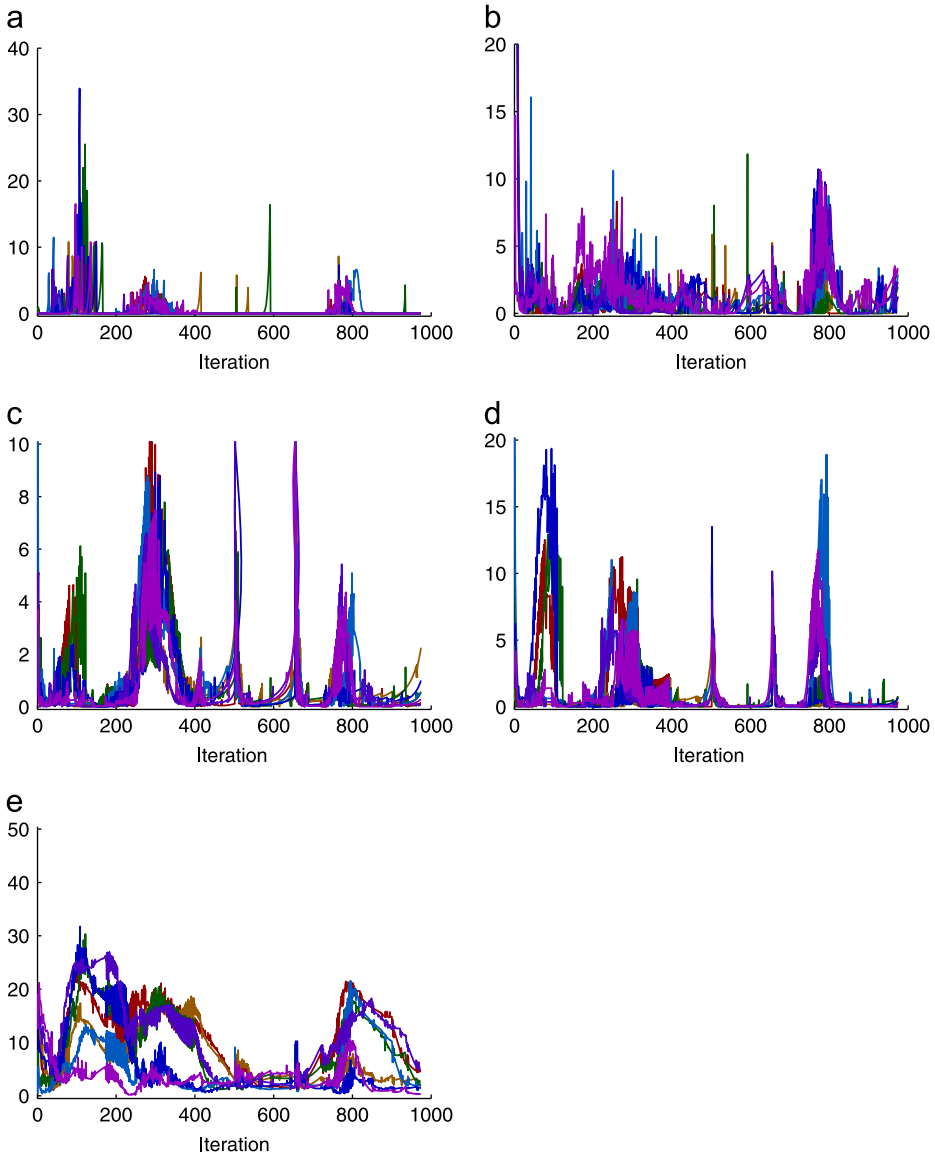

Fig. 10. Evolution of various costs during the mission. (a) Obstacle avoidance. (b) Vehicle collision avoidance. (c) Straight line toward the way-point. (d) Move toward the way-point. (e) Group with other vehicles.

in the MPC strategy as it includes the attitude control loop dynamics and saturations.

Initial conditions: The robustness of the approach has been tested in simulations. Using the set of parameters given by Tables 2a– 5d, 200 simulations were conducted using different, randomly chosen, initial positions for vehicles. Table 6 gives the rates of success, collision and loss-of-a-vehicle over the 200 simulations. The success rate of 98.5% and the absence of collision indicate that our approach allows one to safely accomplish the mission in most situations.

Fig. 13 on the other hand, that shows the evolution of the mean time taken to compute the control of each vehicle throughout the mission, emphasises the main advantage of the systematic search: the computation time is constant in all circumstances.

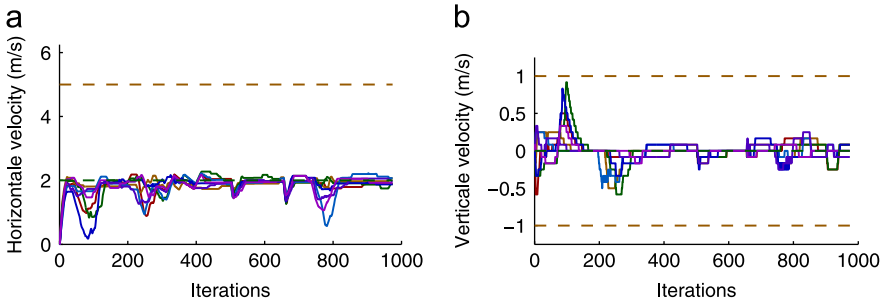

Fig. 11. Velocity of the vehicles. (a) Horizontal velocity. (b) Vertical velocity.

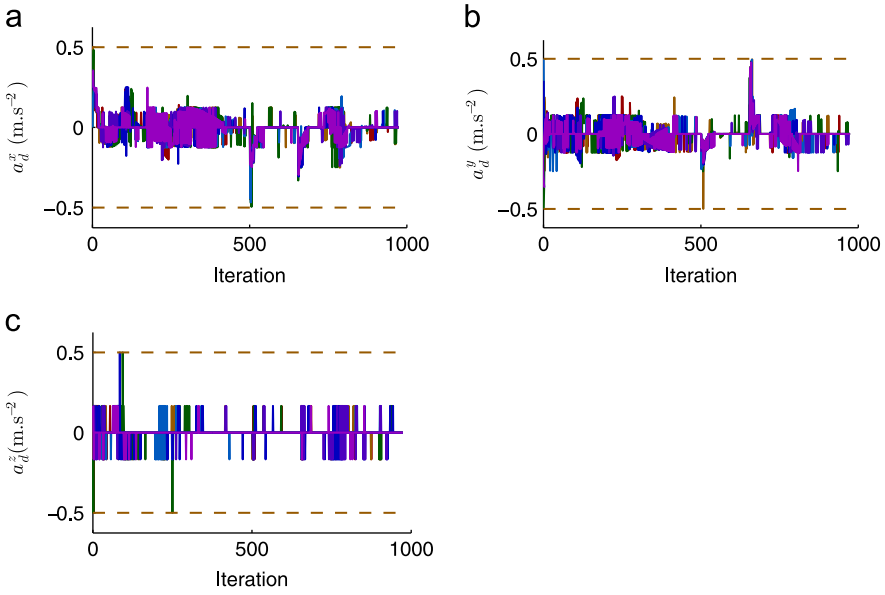

Fig. 12. Evolution of the desired accelerations. (a) Desired acceleration along $x_{\mathcal{F}}$. (b) Desired acceleration along $y_{\mathcal{F}}$. (c) Desired acceleration along $z_{\mathcal{F}}$.

Table 6
Comparison of the systematic search approach with an active-set optimisation method and a combination of active-set initialised with a systematic search.

Optimisation scheme	Systematic search	Active-set	Combination
Success rate (%)	98.5	85	99
Collision rate (%)	0	15	0.5
Vehicle loss rate (%)	1.5	0	0.5
Mean computation time (std) (ms)	18 (1)	102 (16)	121 (18)
Mean mission time (std) (s)	475 (6)	471 (5)	471 (5)
Mean travelled distance (std) (m)	922 (18)	914 (15)	917 (14)
Mean control cost (std)	672 (110)	828 (113)	706 (122)
Mean manoeuvre cost (std)	1658 (155)	1472 (146)	1435 (168)
Mean mission cost (std)	83 451 (8682)	64 719 (5131)	63 933 (5188)
Mean safety cost (std)	13 757 (4959)	12 740 (2658)	12 704 (3911)
Mean total cost (std)	99 538 (10 501)	79 759 (5758)	78 778 (7048)

Comparison with an active-set optimisation: The main objective of the systematic search is to find a feasible, near optimal, control in short, constant, time. In order to evaluate the optimality of the found control, the same 200 simulations were conducted using an active-set approach (function `fmincon` of matlab) instead of the systematic search procedure. Available criteria to evaluate the optimality of the control are given in Table 6, they are the arithmetic mean and standard deviation (std) of the following quantities: time necessary to complete the mission, distance travelled by one vehicle, and overall costs of each of the four categories (control, manoeuvre, mission and safety).

Analysis of comparison: Table 6 shows that results obtained with the active-set approach are significantly worse, in terms of collisions and computation time, than those obtained with systematic search. This is the result of the tendency to fall in local minima that lead to collision.

This weakness can easily be corrected by initialising the optimisation algorithm with a value closer to the optimal one. This can be done using the systematic search to find a good initial value, and then executing the optimisation algorithm. The results obtained with this method are given in Table 6. Although the overall costs are significantly lower, success rate and mission length are very similar to those of systematic search alone.

Influence of search parameters: The influence of the number of control sequence candidates on the performance of the vehicles

has also been studied. This is done by changing one of the following parameters:

N^{dir} : defines the number of candidate values for the argument of the horizontal component of the acceleration in the set of candidate control sequences;

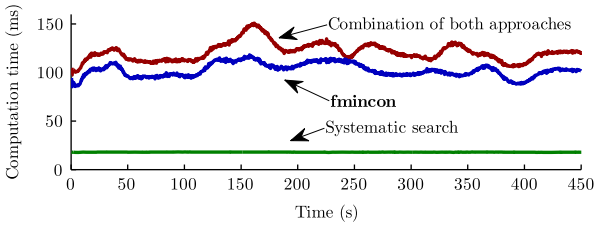

Fig. 13. Mean time taken to compute the control of each vehicle throughout the mission.

N^{norm} : defines the number of candidate values for the norm of the horizontal component of the acceleration in the set of candidates;

N^{z} : defines the number of candidate values for the vertical acceleration in the set of candidates.

These results, given in Tables 7–9, allow one to make two observations. Firstly, and predictably, the control-computation time increases with the number of candidates, as the computation load increases. Secondly, the success rate is penalised when the number of candidates decreases, as the vehicles dispose of a less precise control. This effect is particularly significant for the number of vertical accelerations because of the tight constraints on the altitude of the vehicles that require precise movements along the vertical axis.

Table 7
Influence of N^{dir}.

N^{dir}	4	8	16
Success rate (%)	57	98.5	98.5
Collision rate (%)	0.5	0	1.5
Vehicle loss rate (%)	42.5	1.5	0
Mean computation time (std) (ms)	13 (1)	18 (1)	32 (4)
Mean total cost (std)	109 509 (14 258)	99 538 (10 501)	95 562 (9386)

Table 8
Influence of N^{norm}

N^{norm}	3	4	5
Success rate (%)	95.5	98.5	99
Collision rate (%)	0.5	0	0
Vehicle loss rate (%)	4	1.5	1
Mean computation time (std) (ms)	13 (1)	18 (1)	25 (3)
Mean total cost (std)	130 028 (11 859)	99 538 (10 501)	93 378 (9623)

Table 9
Influence of N^{z}.

N^{z}	3	5	7
Success rate (%)	59.5	98.5	100
Collision rate (%)	2	0	0
Vehicle loss rate (%)	38.5	1.5	0
Mean computation time (std) (ms)	13 (1)	18 (1)	27 (3)
Mean total cost (std)	154 945 (17 369)	99 538 (10 501)	88 478 (8961)

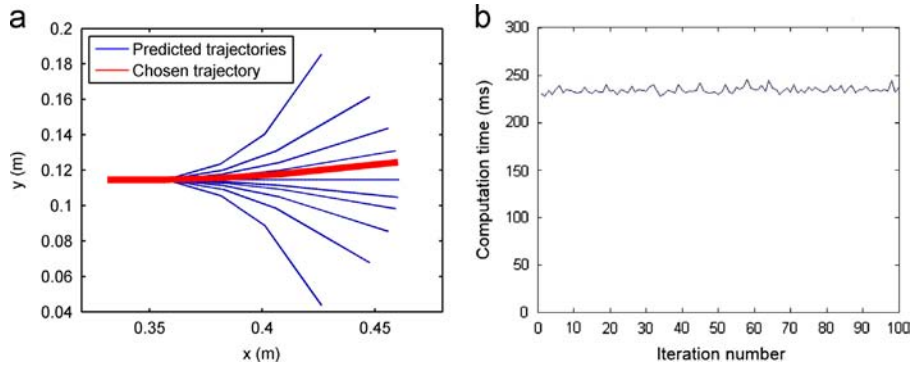

Fig. 14. Illustrative results for the MPC strategy. (a) Example of predicted and chosen trajectories. (b) Computation time at each iteration.

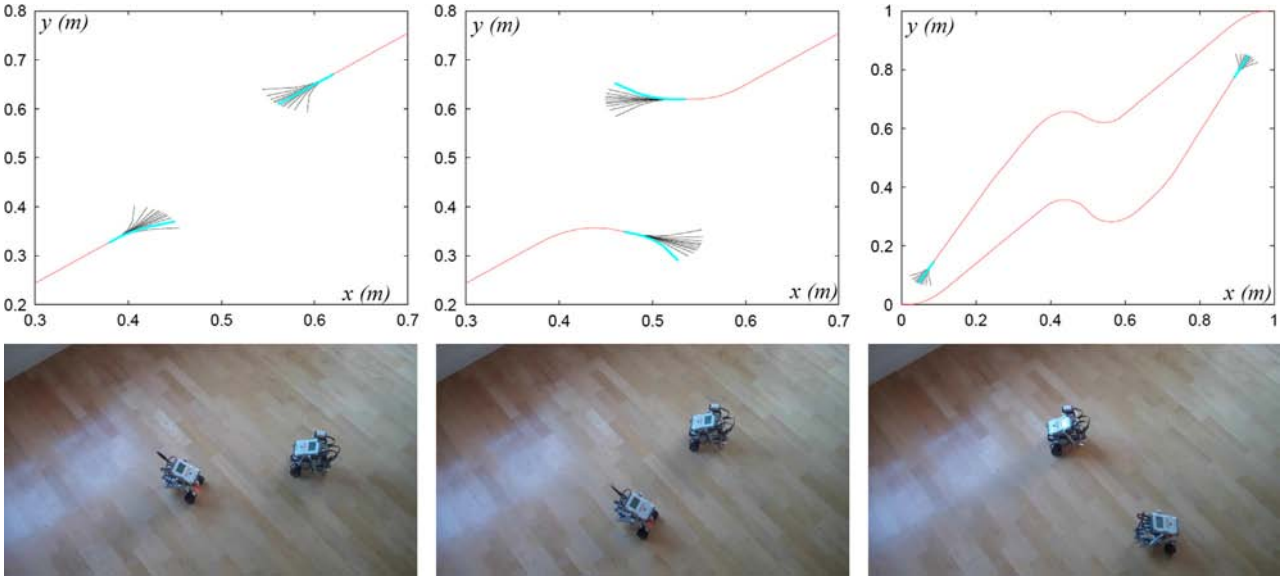

Fig. 15. Collision avoidance trajectory.

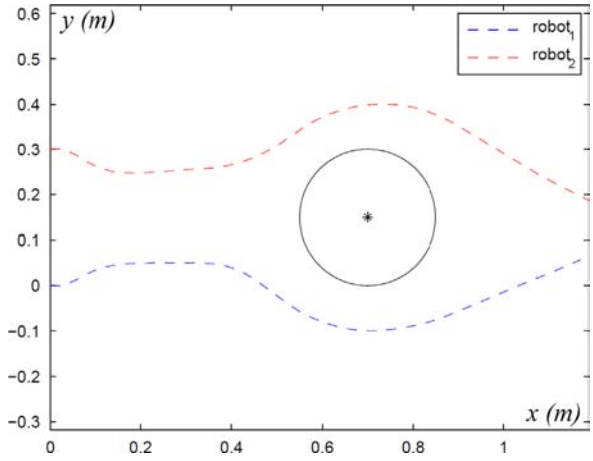

Fig. 16. Fleet navigation with obstacle avoidance (motion from left to right).

8. Experimental results

To go beyond simulation results, experiments on simple ground mobile robots have been set up. The chosen platform is the low-cost LEGO Mindstorms NXT which has been considered in previous works on cooperative multi-robot systems (Benedettelli, Casini, Garulli, Giannitrapani, & Vicino, 2009; Brigandi, Field, & Wang, 2010; Maze, Wan, Namuduri, & Varanasi, 2012; Valente, Hossain, Gronbak, Hallenborg, & Reis, 2010) but not with guidance laws based on MPC, which is usually difficult to embed due to computational load. The chosen discretisation strategy to select the optimal control input (Section 6) makes it possible to embed the MPC guidance laws on such mobile robots with limited capabilities, which is otherwise usually not feasible. An illustration of the resulting trajectory set for the considered case of LEGO Mindstorms NXT robots is given in Fig. 14, where it can also be seen that the computation time required at each iteration is constant and comfortably smaller than Δt (chosen equal to 0.3 s). Also note that the selection algorithm can easily be parallelised on future embedded multi-core architectures, so as to sample more trajectories within the limit of a time step.

Two scenarios involving two robots were considered: collision avoidance and fleet navigation with collision and obstacle avoidance. For the first scenario, the obtained trajectories are given in Fig. 15, where it can be seen that the two robots reach their destination with good accuracy and that collision avoidance is effective. For the second scenario, the obtained trajectories (Fig. 16) illustrate the fleet behaviour of the vehicles (distance $d^v_{\text{des}} = 0.2$ m is respected) before they encounter the obstacle and avoid it (with a safety distance $d^v_{\text{saf}} = 0.1$ m) and finally head toward the same way-point.

9. Conclusion

In this paper the design of a guidance law for the distributed control of a group of cooperative vehicles has been presented. The proposed method consists of the resolution, by each vehicle, of an MPC problem where the cost function is composed of different components dealing with the different objectives of the mission: travel as a flock, reach way-points, avoid obstacles and other vehicles of the group. Intentions of all vehicles, in terms of predicted trajectories, are considered to make the group cooperate. Constraints on vehicles' control inputs and velocities are also handled by the proposed approach.

Since a distributed strategy has been chosen, a systematic search approach has been proposed instead of classical optimisation to ensure that each vehicle can solve its MPC problem efficiently. It has been shown that the resulting computation delay is indeed significantly shorter than with a classical optimisation and constant in all situations without penalising efficiency. These properties make this algorithm suitable for embedded control.

Numerical simulations have been presented in the case of a flock of quadrotor vehicles, and completed by experimental results on ground mobile robots, to illustrate the good performances of the proposed approach.

Future work will focus on a more extensive study of the effects of the search procedure parameters and adding other features in the algorithm such as cooperative area exploration capability with distributed sensors embedded on the vehicles. Another improvement of the proposed algorithm will consist in handling more realistic communication characteristics of the vehicles (*e.g.* limited range and/or direction, communication delays and loss of communication).

Acknowledgements

The authors would like to thank Julien Marzat who contributed to experimental results.

References

Alamir, M., & Murilo, A. (2008). Swing-up and stabilization of a twin-pendulum under state and control constraints by a fast NMPC scheme. *Automatica*, *44*, 1319–1324.

Benedettelli, D., Casini, M., Garulli, A., Giannitrapani, A., & Vicino, A. (2009). A LEGO Mindstorms experimental setup for multi-agent systems. In *Proceedings of the IEEE multi-conference on systems and control* (pp. 1230–1235), Saint Petersburg, Russia.

Bertrand, S., Piet-Lahanier, H., & Hamel, T. (2007). Contractive model predictive control of an unmanned aerial vehicle model. In *IFAC symposium on automatic control in aerospace*.

Brigandi, S., Field, J., & Wang, Y. (2010). A LEGO Mindstorms NXT based multirobot system. In *Proceedings of the IEEE/ASME international conference on advanced intelligent mechatronics* (pp. 135–139), Montreal, Canada.

Dunbar, W., & Murray, R. (2004). Receding horizon control of multi-vehicle formations: A distributed implementation. In *Conference on decision and control* (Vol. 2, pp. 1995–2002).

Dunbar, W. B. (2001). *Model predictive control: Extension to coordinated multi-vehicle formations and real-time implementation*. Technical Report. California institute of technology.

Frew, E. W. (2006). *Receding horizon control using random search for UAV navigation with passive, non-cooperative sensing*. Technical Report. University of Colorado at Boulder.

Kim, H., Shim, D., & Sastry, S. (2002). Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In *American control conference* (Vol. 5, pp. 3576–3581).

Kim, H. J., & Shim, D. H. (2003). A flight control system for aerial robots: *Algorithms and experiments*. *Control Engineering Practice*, *11*, 1389–1400.

Maze, N., Wan, Y., Namuduri, K., & Varanasi, M. (2012). A Lego Mindstorms NXT-based test bench for cohesive distributed multi-agent exploratory systems: Mobility and coordination. In *Proceedings of the AIAA Infotech@Aerospace*, Garden Crove, CA.

Mokhtari, A., & Benallegue, A. (2004). Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle. In *IEEE international conference on robotics and automation* (Vol. 3, pp. 2359–2366).

Müller, M. A., Reble, M., & Allgöwer, F. (2011). A general distributed MPC framework for cooperative control. In *IFAC World congress*.

Olfati-Saber, R., Dunbar, W., & Murray, R. (2003). Cooperative control of multi-vehicle systems using cost graphs and optimization. In *American control conference* (Vol. 3, pp. 2217–2222).

Rochefort, Y., Piet-Lahanier, H., Bertrand, S., Beauvois, D., & Dumur, D. (2011). Guidance of flocks of vehicles using virtual signposts. In *IFAC World congress*.

Rochefort, Y., Piet-Lahanier, H., Bertrand, S., Beauvois, D., & Dumur, D. (2012). Cooperative nonlinear model predictive control for flocks of vehicles. In *Workshop on embedded guidance, navigation and control in aerospace*.

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

Siva, E., & Maciejowski, J. M. (2011). Robust multiplexed MPC for distributed multi-agent systems. In *IFAC World congress*.

Valente, P., Hossain, S., Gronbak, B., Hallenborg, K., & Reis, L. P. (2010). A multi-agent framework for coordination of intelligent assistive technologies. In *Proceedings of the 5th Iberian conference on information systems and technologies*, Santiago de Compostela, Spain.

Venkat, A., Rawlings, J., & Wright, S. (2005). Stability and optimality of distributed model predictive control. In *Conference on decision and control*.

Wang, X., Yadav, V., & Balakrishnan, S. N. (2007). Cooperative UAV formation flying with obstacle/collision avoidance. *IEEE Transactions on Control Systems Technology*, *15*, 672–679.