Robotics and Autonomous Systems 57 (2009) 1094–1106

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

# Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: A distributed receding horizon approach☆

Michael Defoort [a,*], Annemarie Kokosy [b], Thierry Floquet [c], Wilfrid Perruquetti [c], Jorge Palos [b]

[a] LAMIH (UMR-CNRS 8530), Valenciennes University, 59313 Valenciennes, France

[b] LAGIS (UMR-CNRS 8146), ISEN, 41 Boulevard Vauban, 59046 Lille, France

[c] LAGIS (UMR-CNRS 8146), Ecole centrale de Lille, BP 48, Cité Scientifique, and INRIA Lille-Nord Europe, 59651 Villeneuve d'Ascq, France

ARTICLE INFO

*Article history:*
Received 14 December 2007
Received in revised form
15 June 2009
Accepted 8 July 2009
Available online 21 July 2009

*Keywords:*
Decentralized intelligence
Receding horizon planning
Nonholonomic mobile robot

ABSTRACT

This paper presents a decentralized motion planner for a team of nonholonomic mobile robots subject to constraints imposed by sensors and the communication network. The motion planning scheme consists of decentralized receding horizon planners that reside on each vehicle to achieve coordination among flocking agents. The advantage of the proposed algorithm is that each vehicle only requires local knowledge of its neighboring vehicles. The main requirement for designing an optimal conflict-free trajectory in a decentralized way is that each robot does not deviate too far from its presumed trajectory designed without taking the coupling constraints into account. A comparative study between the proposed algorithm and other existing algorithms is provided in order to show the advantages, especially in terms of computing time. Finally, experiments are performed on a team of three mobile robots to demonstrate the validity of the proposed approach.



## 1. Introduction

The research effort in multi-robot system relies on the fact that multiple robots have the possibility to solve problems more efficiently than a single one. Cooperative control gives rise to significant theoretical challenges and has various engineering applications including manufacturing [1,2], surveillance [3,4], and space exploration [5,6].

In this paper, the problem of interest is the navigation of cooperative nonholonomic mobile robots evolving in environments with obstacles. Here, the vehicles are dynamically decoupled but have common constraints that make them interact. Indeed, each robot has to avoid collision with the other ones. Moreover, since cooperative algorithms involve information exchange between robots [7,8], some communication links must be maintained during the movement. Because of the limited range of transmitters and receivers, the distance between two vehicles which may exchange information will naturally be constrained. Besides maintaining the communication, the feasibility of the trajectories implies respect of the dynamic constraints, as well as avoiding obstacles and collisions. In this paper, we will focus solely on these dynamic and geometric aspects and ignore mobile networking factors, such as fading, cross talk, and delay, which can also affect the quality of communication between the vehicles.

Motion planning consists in generating a collision-free trajectory from the initial to the final desired positions for each vehicle. Depending on the distance that the robots have to travel, the computation of complete trajectories from start until finish may be computationally expensive. Moreover, the field of view of the physical sensors is limited. Therefore, the trajectories have to be computed gradually over time while the mission unfolds. This can be accomplished using an on-line receding horizon planner, in which partial trajectories from an initial state toward the goal are computed by solving an optimal control problem over a limited horizon [9].

Cooperative approaches for motion planning can be further broken down into centralized and decentralized (distributed) methods. The centralized approach is usually based on the resolution of an optimal control problem via a game theoretic framework [10], semi-definite programming [11] or non-linear programming [12]. However, its computation time, which scales exponentially with the number of vehicles, its communication requirement and its lack of security make it prohibitive. This is true even when the most advanced optimization solvers and much simplified linear vehicle dynamics are used [13]. The main challenge is to formulate a simple decentralized strategy which results in a team behavior similar to what is obtained with a centralized approach.

Some decentralized algorithms are based on a receding horizon approach. In [14], the cost function of the optimal control problems

☆ This work was partially supported by the JSPS Postdoctoral Fellowship and the FEDER (European Funds of Regional Development).

* Corresponding author.
*E-mail address:* michael.defoort@gmail.com (M. Defoort).

0921-8890/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2009.07.004

reflects the communication constraints between robots. Then, a distributed receding horizon implementation is provided for unconstrained subsystems (decoupled input constraints only). In [15], the authors propose a strategy where each robot optimizes locally for itself as well as for every neighbor at each update, resulting in an increase in the computing time and a decrease in the decentralization. Furthermore, in order to ensure collision avoidance, some emergency strategies are defined [16]. In [17,18], the decentralized receding horizon planner is solved using mixed-integer linear programming (MILP). Every robot is allotted a time slot in which to compute its own dynamically feasible collision-free trajectory. An extension of this approach is proposed in [19]. This scheme, based on a leader–follower architecture, creates a queueing order of non-conflicting groups of vehicles, where each group updates the trajectory sequentially and communicates the solution to its neighbors. One advantage of this approach is that it is easy to implement. However, there is no explicit feedback from the follower to the leader. Another disadvantage is that the leader is a single point of failure.

Other decentralized strategies, which can be computationally fast, include rule-based approaches [20], probabilistic approaches [21,22] and artificial potential fields [23–26]. However, most of these techniques have been successfully developed for oversimplified vehicle dynamics without explicitly taking constraints into account. For instance, collision avoidance constraints are usually translated into minimization of functions whose value becomes very high when a direction which leads to collisions is chosen. Such an approach does not account for real-life issues like limited speed, acceleration and actuation authority of the robot.

The scheme proposed in this paper attempts to overcome these limitations and other inherent difficulties associated with achieving navigation for constrained nonholonomic mobile robots via a decentralized receding horizon approach. Each robot optimizes only for its own trajectory at each update. The approach is cooperative since every neighboring vehicle exchanges its intentions via wireless communication prior to each update. The intention of each robot is a presumed trajectory designed without taking the coupling constraints between neighboring vehicles into account. Providing that the robot does not deviate too far from its presumed trajectory, a conflict-free trajectory satisfying all the constraints can be found. Furthermore, the fully distributed algorithm scales well with respect to the number of robots. Contrary to most of the existing motion planners, the distributed planning algorithm guarantees the feasibility of the vehicle path, which is crucial for real time implementation.

The outline of this paper is as follows. In Section 2, the problem setup is described. Section 3 introduces the decentralized motion planning approach which only uses local information. In Section 4, a comparative study between the proposed algorithm and other existing algorithms is provided in order to show the advantages, especially in terms of computing time. Section 5 shows the results of the implementation on a real system of autonomous robots, while Section 6 provides a conclusion and future prospects.

## 2. Problem setup

### 2.1. Modeling of mobile robot

Each mobile robot $R_n$ ($n \in \{1, \dots, N\}$), shown in Fig. 1, is of unicycle type. The $n$th robot body is of symmetric shape and the center of mass is at the geometric center $C_n$ of the body. It has two driving wheels fixed to the axis which passes through $C_n$ and one passive centered orientable wheel. The two fixed wheels, separated by $2\rho_n$, are independently controlled by two actuators (DC motors) and the passive wheel prevents the robot from tipping over as it moves on a plane. In this paper, we assume that the

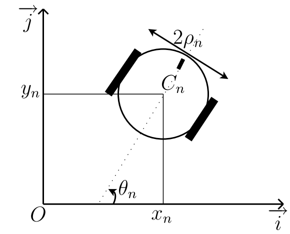



**Fig. 1.** Unicycle-type mobile robot.

motion of the passive wheel can be ignored in the dynamics of the mobile robot. The center of mass $C_n$, whose coordinates are $(x_n, y_n)$, is located at the intersection of a straight line passing through the middle of the vehicle and the axis of the two driving wheels. The configuration of the robot can be described by

$$q_n(t) = [x_n(t), y_n(t), \theta_n(t)]^{\mathrm{T}},$$

where $\theta_n(t)$ is its orientation in the global frame.

In this paper, the kinematics of a wheeled-mobile robot is shown under the nonholonomic constraints (see [27] for details). The pure rolling and nonslipping nonholonomic conditions are described by

$$A^{\mathrm{T}}(q_n)\dot{q}_n = 0 \quad \text{with} \quad A^{\mathrm{T}}(q_n) = \begin{bmatrix} -\sin\theta_n & \cos\theta_n & 0 \end{bmatrix}.$$

The kinematic equations can be written as follows:

$$\dot{q}_n(t) = f(q_n(t), u_n(t)), \tag{1}$$

where the vector field $f : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$ and control inputs $u_n$ are defined as

$$\begin{cases} f(q_n(t), u_n(t)) = \begin{bmatrix} \cos\theta_n(t) & 0 \\ \sin\theta_n(t) & 0 \\ 0 & 1 \end{bmatrix} u_n(t), \\ u_n(t) = [v_n(t), w_n(t)]^{\mathrm{T}}. \end{cases}$$

$v_n(t)$ and $w_n(t)$ are the linear and angular velocities, respectively. System (1) is constrained to take the practical limitations on the vehicle velocities into account, i.e. $\forall t \geq 0$,

$$\begin{cases} |v_n(t)| \leq v_{n,max}, \\ |w_n(t)| \leq w_{n,max}. \end{cases} \tag{2}$$

### 2.2. Problem formulation

The following assumptions are made in this study:

(1) The vehicle team is composed of $N$ nonholonomic mobile robots.
(2) Each robot $R_n$ ($n \in \{1, \dots, N\}$) knows its initial configuration $q_n(t_{ini})$ ($t_{ini}$ is the initial time instant) and its goal configuration $q_n(t_{fin}) = q_{n,fin}$ ($t_{fin}$ is the final time instant). Note that the task location $q_{n,fin}$ is chosen by some kind of high-level directive, for instance, the output of a task allocation problem [28].
(3) Each mobile robot has a physical safety area, represented by a 2-D circle of center $C_n = (x_n, y_n)$ and of radius $\rho_n$.
(4) Robots have on-board sensors which can detect surrounding objects and vehicles within a range with a small margin of error,
(5) The obstacles are static and convex. Without loss of any generality,[1] the $i$th ($i = 1, \dots$) obstacle in the environment is included in a 2-D circle, denoted by $\mathcal{B}_i(O_i, r_i)$, where $O_i = (X_i, Y_i)$ is its center and $r_i$ is its radius.

[1] It is trivial to allow the envelope of an obstacle to be represented by union/intersection of several circles. The envelopes could also be polygonal. Mathematically, circular envelopes can be represented by second-order inequalities while polygonal envelopes can be described by first-order linear inequalities.

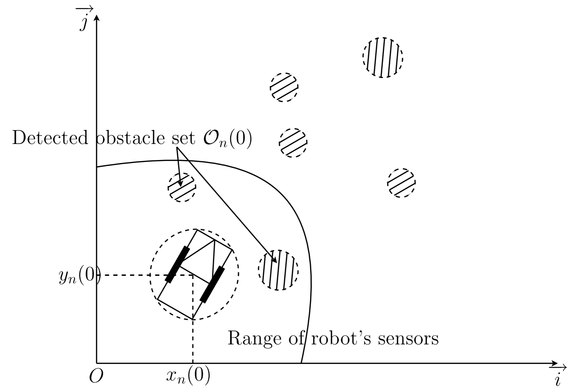



**Fig. 2.** Detected obstacle set at time $t_k = 0$ s.

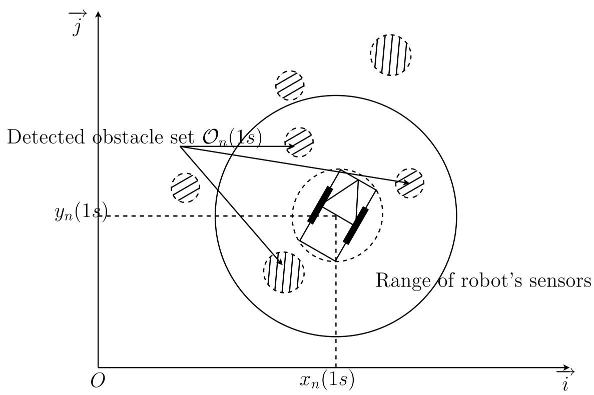



**Fig. 3.** Detected obstacle set at time $t_k = 1$ s.

(6) Each robot $R_n$ can reliably communicate with another robot $R_p$ ($p \neq n, p \in \{1, \dots, N\}$) of the team. Its broadcasting range $d_{n,com} \in \mathbb{R}^+$ is strictly larger than $\rho_n + \rho_p$.

Let us consider the following distributed integrated cost for multi-vehicle navigation:

$$L(q, u) = \sum_{n=1}^{N} L_n(q_n, u_n, q_{n,fin}), \tag{3}$$

where $L_n(q_n, u_n, q_{n,fin}) : \mathbb{R}^3 \times \mathbb{R}^2 \times \mathbb{R}^3 \longmapsto \mathbb{R}^+$ is the integrated cost for one robot stabilization (it is assumed that $L_n(q_n, u_n, q_{n,fin}) = 0$ if and only if $q_n = q_{n,fin}$ and $u_n = 0$), $q = \left[q_1^T, \dots, q_N^T\right]^T$ and $u = \left[u_1^T, \dots, u_N^T\right]^T$ are the concatenated trajectory and control input.

**Definition 1.** $\forall t_k \in \left[t_{ini}, t_{fin}\right]$, $\forall R_n$ ($n \in \{1, \dots, N\}$), the detected obstacle set $\mathcal{O}_n(t_k)$ is defined as the subset $\mathcal{O}_n(t_k) \subset \{\mathcal{B}_1(O_1, r_1), \mathcal{B}_2(O_2, r_2), \dots, \}$ of $M_n$ obstacles in the range of the robot sensors.

Note that the detected obstacle set is time dependent and evolves as long as the robot moves and discovers new obstacles (see Figs. 2 and 3). In order to ensure the collision avoidance with obstacles, for all $t \geq 0$, the distance between the robot and the detected obstacles (i.e. $O_{m_n} \in \mathcal{O}_n(t)$), $d(q_n(t), O_{m_n}) = \sqrt{(x_n(t) - X_{m_n})^2 + (y_n(t) - Y_{m_n})^2}$ must satisfy

$$d(q_n(t), O_{m_n}) \geq \rho_n + r_{m_n}. \tag{4}$$

The robots should also remain at a safe distance (i.e. $\rho_n + \rho_p$) from each other to avoid collisions. This coupling constraint can be expressed as follows: For each pair $(R_n, R_p) \in \mathcal{R} \times \mathcal{R}$, $n \neq p$,

$$d(q_n(t), q_p(t)) \geq \rho_n + \rho_p. \tag{5}$$

Similar to [24,25], we consider a coordinated motion planning problem, where besides avoiding collisions, robots need to cooperate and maintain formation constraints in order to reach their goals. Indeed, the positions of the robots, the requirements of the desired task and the limited range of transmitters and receivers dictate the topology of the network. That is to say, the communication links between some vehicles must be maintained during the movement. This network can be expressed by a communication graph, defined as follows.

**Definition 2.** A *communication graph* $(\mathcal{R}, \mathcal{E}, \mathcal{S})$ is an undirected labeled graph consisting of

- a set $\mathcal{R} = \{R_1, \dots, R_N\}$ of $N$ nodes representing the robots;
- a set of edges $\mathcal{E} \subseteq \mathcal{R} \times \mathcal{R}$ encoding the communication links; and
- a set of edge constraints $\mathcal{S}$ describing the condition under which each edge is maintained.

For each element of $\mathcal{E}$, there is one corresponding element in $\mathcal{S}$. For instance, if the task requires cooperation between robots $R_n$ and $R_p$, the communication link is represented by the edge $(R_n, R_p) \in \mathcal{E}$ and the associated constraint due to the limited range of transmitters and receivers. This constraint can be written as

$$d(q_n(t), q_p(t)) \leq \min(d_{n,com}, d_{p,com}), \tag{6}$$

where $d(q_n(t), q_p(t)) = \sqrt{(x_n(t) - x_p(t))^2 + (y_n(t) - y_p(t))^2}$ is the distance between robots $R_n$ and $R_p$.

**Remark 1.** The construction of the communication graph, which depends on the task the robots are performing, is beyond the scope of this paper. A methodology based on a leader election algorithm is proposed in [29]. Note that the construction of this graph is done only at the beginning of the mission, or when it is necessary during the execution of the task (e.g. one of the robots fails or a new robot is incorporated).

The objective is to compute the optimal trajectory for each vehicle, from $q_n(t_{ini})$ to $q_{n,fin}$, which simultaneously enforces nonholonomic constraints (1)–(2), avoidance constraints (4)–(5) and formation constraints (6) outlined above, in a decentralized cooperative way and according to the performance criteria (3).

## 3. Decentralized motion planning scheme

As discussed in Section 1, the computation of complete trajectories from start until finish is infeasible. Therefore, the trajectory has to be computed gradually over time while the mission unfolds. It can be accomplished using an on-line receding horizon planner, in which partial trajectories from an initial state toward the goal are computed by solving a constrained finite time optimal control problem over a limited horizon.

In this section, we propose a way to decentralize the receding horizon planning problem. In this approach, each vehicle solves a reduced subproblem to determine its control inputs. In every distributed optimal control problem, the same constant planning horizon $T_p \in \mathbb{R}^+$ and constant update period $T_c \in \mathbb{R}^+$ ($T_c < T_p$) are used. In practice, $T_c$ is typically the time allocated for the resolution of the distributed optimal control problem. At each update, denoted $\tau_k$ ($k \in \mathbb{N}$),

$$\tau_k = t_{ini} + kT_c, \tag{7}$$

each robot computes, in parallel, an optimal planned trajectory satisfying constraints (1), (2) and (4)–(6). The distribution is achieved by having each vehicle exchange its presumed information with other vehicles. A key element of this work is that the vehicles must only exchange information with robots that could have direct conflicts (i.e. may produce a collision or may lose communication), enabling the local optimization to be based on local information. This is important because it reduces the communication requirements and the computational complexity.

### 3.1. Conflict sets

For each vehicle $R_n$, let us define two subsets of robots characterizing conflicts that may occur.

- The inter-robot collision conflict set $\mathcal{C}_{n,collision}(\tau_k) \subset \mathcal{R}$ is the subset of all vehicles $R_{p\neq n}$ for which the separation distance verifies

$$d(q_n(\tau_k), q_p(\tau_k)) \leq \rho_n + \rho_p + (v_{n,max} + v_{p,max})(T_p + T_c). \quad (8)$$

- The communication conflict set $\mathcal{C}_{n,com}(\tau_k) \subset \mathcal{R}$ is the subset of all vehicles $R_{p\neq n}$ such that $(R_n, R_p) \in \mathcal{E}$ and for which the separation distance verifies

$$d(q_n(\tau_k), q_p(\tau_k)) \geq \min(d_{n,com}, d_{p,com}) - (v_{n,max} + v_{p,max}) \times (T_p + T_c). \quad (9)$$

Once these two subsets are defined, the conflict set can be expressed as

$$\mathcal{C}_n(\tau_k) = \mathcal{C}_{n,collision}(\tau_k) \bigcup \mathcal{C}_{n,com}(\tau_k).$$

**Remark 2.** For all $(R_n, R_p) \in \mathcal{R} \times \mathcal{R}$ with $R_p \notin \mathcal{C}_n(\tau_k)$, the coupling constraints (5)–(6) between robots $R_n$ and $R_p$ are satisfied during the interval $[\tau_k, \tau_{k+1} + T_p]$.

### 3.2. Decentralized motion planning algorithm

Let us denote the obstacle detection horizon as $T_d \in \mathbb{R}^+(T_p < T_d)$, which depends on the range of the robot sensors. Over any interval $[\tau_k, \tau_k+T_d]$, every vehicle $R_n$ must presume some preferred trajectories for robots $R_p$ belonging to the conflict set in order to plan its optimal conflict-free trajectory. Two difficulties can be stated:

- the definition of a unique presumed trajectory for each robot,
- the coherence between what a vehicle plans to do (the real optimal planned trajectory) and what the other robots believe that the vehicle will plan to do (the presumed trajectory).

The proposed solution is to divide each receding horizon planning problem into two steps. In each time interval $[\tau_{k-1}, \tau_k)$:

**Step 1.** Each robot $R_n$ computes its **presumed trajectory**, denoted by $\widehat{q}_n(t, \tau_k)$ and its corresponding control input, denoted by $\widehat{u}_n(t, \tau_k)$. This trajectory is obtained without taking the coupling constraints (5)–(6) into account. It is evaluated over the obstacle detection horizon $T_d$, i.e. $t \in [\tau_k, \tau_k + T_d]$.
**Step 2.** Robots which may produce a collision or may lose communication adjust their presumed trajectory by taking constraints (5)–(6) into account and using local exchanged information. This new trajectory, denoted by $q_n^*(t, \tau_k)$, is called the **optimal planned trajectory** and is evaluated over the planning horizon $T_p$, i.e. $t \in \left[\tau_k, \tau_k + T_p\right]$. It is the trajectory that the robot must track during the time interval $[\tau_k, \tau_{k+1}]$. The associated control inputs are denoted by $u_n^*(t, \tau_k)$.

**Remark 3.** By design, the presumed trajectory is the same in every distributed receding horizon planning problem in which it occurs, i.e. $\forall R_p \in \mathcal{C}_n(\tau_k)$, the same trajectory is presumed for $R_n$.

A more formal description follows. Over each interval $[\tau_{k-1}, \tau_k)$, let the following optimal control problem $\widehat{P}_n(\tau_k)$ be associated with the $n$th robot which consists in determining the presumed control input $\widehat{u}_n(t, \tau_k)$ and the presumed trajectory $\widehat{q}_n(t, \tau_k)$ which only satisfies constraints (1)–(4):

**Problem** $\widehat{P}_n(\tau_k)$:

$$\min_{\widehat{q}_n(t,\tau_k),\widehat{u}_n(t,\tau_k)} \int_{\tau_k}^{\tau_k+T_d} L_n(\widehat{q}_n(t, \tau_k), \widehat{u}_n(t, \tau_k), q_{n,fin})\mathrm{d}t, \quad (10)$$

subject to: $\forall t \in [\tau_k, \tau_k + T_d]$,

$$\begin{cases} \dot{\widehat{q}}_n(t, \tau_k) = f(\widehat{q}_n(t, \tau_k), \widehat{u}_n(t, \tau_k)), \\ \widehat{q}_n(\tau_k, \tau_k) = q_n^*(\tau_k, \tau_{k-1}), \\ \widehat{u}_n(\tau_k, \tau_k) = u_n^*(\tau_k, \tau_{k-1}), \\ |\widehat{v}_n(t, \tau_k)| \leq v_{n,max}, \\ |\widehat{w}_n(t, \tau_k)| \leq w_{n,max}, \\ d(\widehat{q}_n(t, \tau_k), O_{m_n}) \geq \rho_n + r_{m_n}, \quad \forall O_{m_n} \in \mathcal{O}_n(\tau_{k-1}). \end{cases} \quad (11)$$

**Remark 4.** During the initialization step, that is to say before the robots move, we denote

$$\begin{cases} \tau_{-1} = t_{ini} \\ q_n^*(\tau_0, \tau_{-1}) = q_n(t_{ini}) \\ u_n^*(\tau_0, \tau_{-1}) = u_n(t_{ini}). \end{cases}$$

Given the conflict sets $\mathcal{C}_{n,collision}(\tau_{k-1})$ and $\mathcal{C}_{n,com}(\tau_{k-1})$, the presumed trajectory $\widehat{q}_n(t, \tau_k)$ and those of its neighbors, let us define the optimal control problem $P_n^*(\tau_k)$ associated with the $n$th robot, over each interval $[\tau_{k-1}, \tau_k)$, which consists in determining the optimal control input $u_n^*(t, \tau_k)$ and the optimal predicted trajectory $q_n^*(t, \tau_k)$ which satisfies all the constraints (1)–(6):

**Problem** $P_n^*(\tau_k)$:

$$\min_{q_n^*(t,\tau_k),u_n^*(t,\tau_k)} \int_{\tau_k}^{\tau_k+T_p} L_n(q_n^*(t, \tau_k), u_n^*(t, \tau_k), q_{n,fin})\mathrm{d}t, \quad (12)$$

subject to: $\forall t \in \left[\tau_k, \tau_k + T_p\right]$,

$$\dot{q}_n^*(t, \tau_k) = f(q_n^*(t, \tau_k), u_n^*(t, \tau_k)), \quad (13)$$

$$q_n^*(\tau_k, \tau_k) = q_n^*(\tau_k, \tau_{k-1}), \quad (14)$$

$$u_n^*(\tau_k, \tau_k) = u_n^*(\tau_k, \tau_{k-1}), \quad (15)$$

$$|v_n^*(t, \tau_k)| \leq v_{n,max}, \quad (16)$$

$$|w_n^*(t, \tau_k)| \leq w_{n,max}, \quad (17)$$

$$d(q_n^*(t, \tau_k), O_{m_n}) \geq \rho_n + r_{m_n}, \quad \forall O_{m_n} \in \mathcal{O}_n(\tau_{k-1}), \quad (18)$$

$$d(q_n^*(t, \tau_k), \widehat{q}_p(t, \tau_k)) \geq \rho_n + \rho_p + \xi, \quad \forall R_p \in \mathcal{C}_{n,collision}(\tau_{k-1}), \quad (19)$$

$$d(q_n^*(t, \tau_k), \widehat{q}_{p'}(t, \tau_k)) \leq \min(d_{n,com}, d_{p',com}) - \xi, \quad \forall R_{p'} \in \mathcal{C}_{n,com}(\tau_{k-1}), \quad (20)$$

$$d(q_n^*(t, \tau_k), \widehat{q}_n(t, \tau_k)) \leq \xi, \quad (21)$$

where $\xi \in \mathbb{R}^+$ is a constant. Here, the optimal planned trajectory $q_n^*(t, \tau_k)$ is constrained to be at most at a distance $\xi$ from the corresponding presumed trajectory $\widehat{q}_n(t, \tau_k)$ (see Eq. (21)). This constraint enforces the degree of correspondence between the optimal trajectory and the presumed trajectory which is the trajectory that other robots rely on. Furthermore, in order to guarantee the collision avoidance between vehicles and the preservation of the communication links, deformation $\xi$ due to the mismatch between the presumed and the optimal planned trajectories is added (see Eqs. (19)–(20)).

The decentralized receding horizon planner is described by the flow chart given in Fig. 5. One can see that during the initialization (i.e. before robot movement), each one only computes its own trajectory using local information and enforcing the correspondence constraint (21). This process is then repeated during the robots' movement, over the interval $[\tau_0, \tau_1)$, and so on, until they reach a neighborhood of their goal $q_{n,fin}$. As such, new information can be taken into account in the next iteration.

**Remark 5.** One can note that constraints (14)–(15) which guarantee the continuity of the trajectory and control inputs need the optimal trajectory $q_n^*(\tau_k, \tau_{k-1})$ and control inputs $u_n^*(\tau_k, \tau_{k-1})$ computed in the previous step. Therefore, in the proposed strategy,

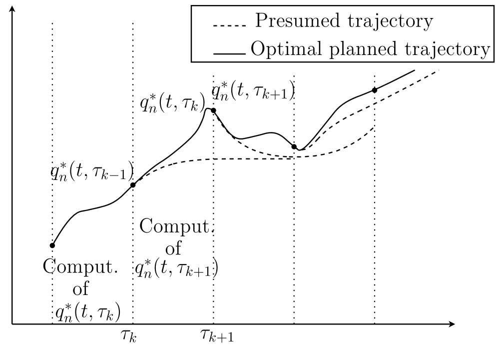



**Fig. 4.** Implementation of the receding horizon motion planner.

the receding horizon planner is not used in order to reject external disturbances or inherent discrepancies between the model and the real process, as it is usually done (see for instance [30]). However, it takes the real time constraint into account. Indeed, each mobile robot has a limited time to compute its optimal trajectory. The time allocated to make its decision depends on its perception sensors, its computation delays, etc., and is less than the update period $T_c$ (see Fig. 4).

Note that the discussed claim for robustness can be effectively accomplished by sliding mode control [31]. Indeed, for real experiments, a low level controller should be added such that each robot tracks its optimal planned trajectory.

**Remark 6.** A compromise must be made between reactivity, optimality and computation time. Indeed, the planning and detection horizon must be sufficiently small in order to have good enough results in terms of computation time. However, the planning horizon $T_p$ must be higher than the update period $T_c$ in order to guarantee reactivity and obstacle avoidance for the next planning problems. Similarly, it must be lower than the detection horizon $T_d$ in order to have sufficient authority to avoid collisions between robots and keep the communication links.

Some specific advantages of the proposed decentralized algorithm are enumerated below:

- security (without any supervisor or leader),
- high level of decentralization (each robot only knows its own trajectory, its own desired goal and a presumed trajectory of vehicles for which conflicts may occur),
- low computation time,
- the real time constraint is not ignored,
- quasi-optimal trajectory,
- good enough communication bandwidth (local information).[2]

### *3.3. Technique for solving the receding horizon planning problems*

There are three components for the real time resolution of optimal control problems $\widehat{P}_n(\tau_k)$ (resp. $P_n^*(\tau_k)$): determination of the flat outputs, B-spline parameterization and constrained feasible sequential quadratic programming.

The key approach is to determine outputs such that Eq. (1) is mapped to a lower-dimensional output space. This will imply that the problem becomes computationally more efficient to solve. Using the flatness property of system (1) (see [32] for further details about flatness), all system variables can be differentially parameterized by $x_n$, $y_n$ as well as a finite number of their time derivatives. Indeed, $\theta_n$, $v_n$ and $w_n$ can be expressed by $x_n$, $y_n$ and their first and second time derivatives, i.e.

$$\theta_n = \arctan\frac{\dot{y}_n}{\dot{x}_n}, \quad v_n = \sqrt{\dot{x}_n^2 + \dot{y}_n^2} \quad \text{and} \quad w_n = \frac{\ddot{y}_n\dot{x}_n - \ddot{x}_n\dot{y}_n}{\dot{x}_n^2 + \dot{y}_n^2}. \quad (22)$$

Once the performance criteria (10) (resp. criteria (12)) and constraints (11) (resp. constraints (13)–(21)) are mapped into the flat output space, the presumed (resp. optimal planned) trajectory is planned in this space (see Fig. 6).

Then, in order to transform the optimal trajectory generation problem into a parameter optimization one, a piecewise polynomial function, the B-spline, is adopted to specify the trajectory. The B-spline functions are chosen as basis functions due to their flexibility and ease of enforcing continuity across breakpoints. The B-spline is a function defined by a series of knots called control knots. In our study, the three-order B-spline basis functions are used to parameterize the trajectory. For problem $\widehat{P}_n(\tau_k)$ (resp. $P_n^*(\tau_k)$), the time interval $[\tau_k, \tau_k + T_d]$ (resp. $\left[\tau_k, \tau_k + T_p\right]$) is divided into $n_{knot}$ equal segments with the corresponding union set of breakpoints:

$$\left[nod_0 = \tau_k < nod_1 < \cdots < nod_{n_{knot}} = \tau_k + T_d \,(\text{resp. } T_p)\right]. \quad (23)$$

Fig. 7 depicts an example of a piecewise polynomial function and the three-order B-spline basis functions associated to the breakpoints [0, 0.25, 0.5, 0.75, 1] (i.e. $n_{knot} = 4$).

The trajectories of the flat outputs are written in terms of finite-dimensional B-spline curves as

$$\begin{bmatrix} x(t,\tau_k) \\ y(t,\tau_k) \end{bmatrix} = \sum_{j=0}^{n_{knot}} C_j B_{j,3}(t), \quad (24)$$

where $C_j \in \mathbb{R}^2$ are the control points and $B_{j,3}$ is the three-order B-spline basis function [33].

Finally, the time domain is truncated into smaller intervals by quadratic laws. The optimal control points $C_j$ are numerically found using the constrained feasible sequential quadratic optimization algorithm [34]. See [35] for a detailed analysis of the efficiency of this approach. To finish, the open-loop control inputs are deduced using Eq. (22).

## 4. Simulation results

In order to highlight the performance of the proposed decentralized strategy, a comparative study with other existing receding horizon motion planners for a team of nonholonomic mobile robots subject to constraints imposed by sensors and the communication network is done. The comparison criteria are: (1) the computation time, (2) the information flow, (3) the amount of exchanged information, (4) the ease of implementation, and (5) the optimal traveling time.

### *4.1. Generalities on existing cooperative receding horizon motion planners*

We have implemented and tested the following approaches suitable for our motion planning problem under the same conditions[3].

[2] Since each trajectory and control input are approximated using B-spline functions, only control knots and optimal control points are exchanged pairwise (see Section 3.3 for further details).

[3] One can note that the existing strategies have been proposed in order to solve the motion planning problem for linear systems. Nevertheless, using the techniques given in Section 3.3 (i.e. flat outputs, B-spline parameterization and constrained feasible sequential quadratic programming), these algorithms can easily be extended to nonlinear systems.

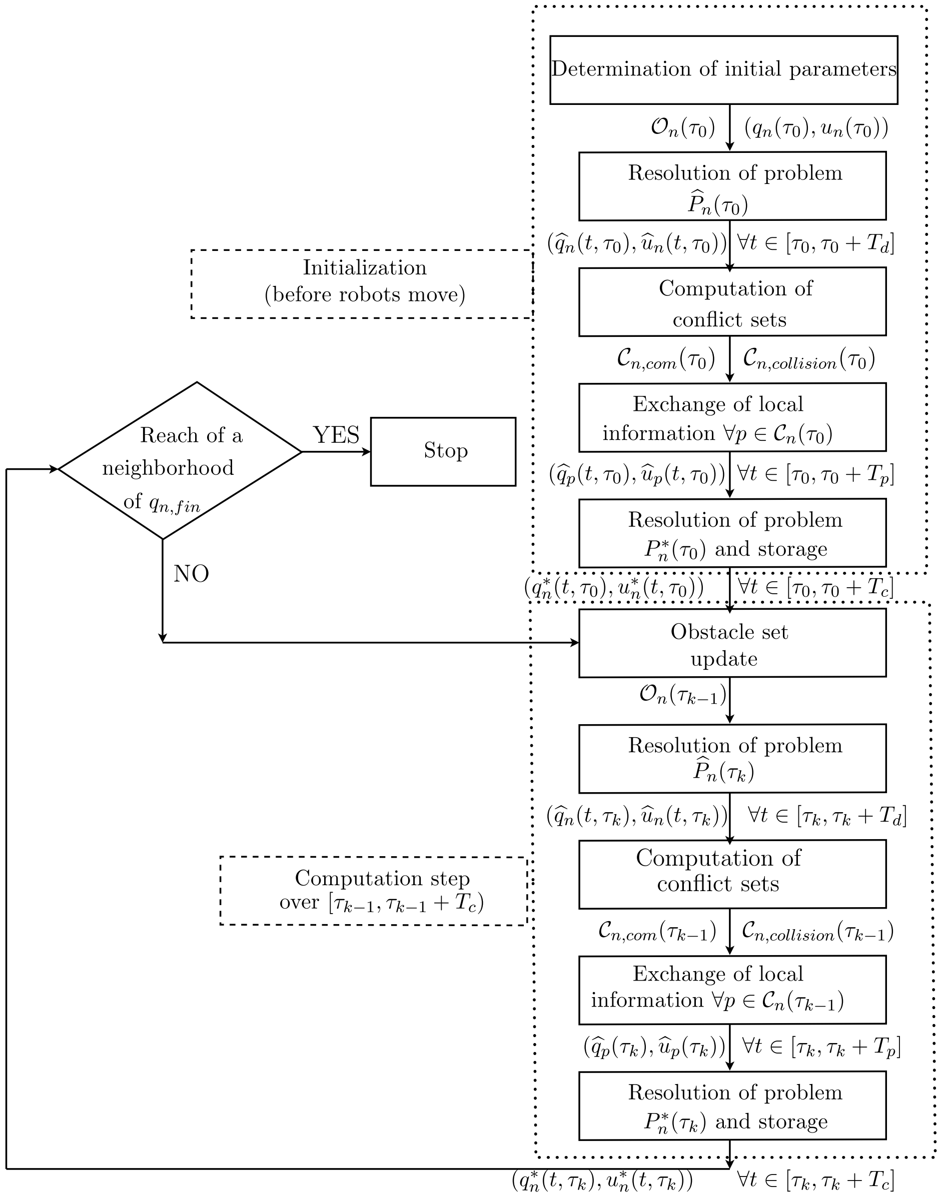



**Fig. 5.** Flow chart of the proposed distributed receding horizon motion planner.

- A centralized algorithm where the trajectories are computed via a supervisor by typically solving a large-dimension scale optimization problem.
- A "weakly" decentralized receding horizon approach [15] where each robot optimizes locally for itself and every neighbor at each update. In this case, over any interval $[\tau_{k-1}, \tau_k)$, only configuration $q_p^*(\tau_k, \tau_{k-1})$ is exchanged pairwise. This fact enables one to decrease the information flow. From $q_p^*(\tau_k, \tau_{k-1})$ $(\forall p \in \mathcal{C}_n(\tau_{k-1}))$, robot $R_n$ generates its optimal planned trajectory $q_n^*(t, \tau_k)$ and the presumed trajectories $\widehat{q}_p(t, \tau_k)$.

  Two problems arise from using such a strategy. On the one hand, the lack of correspondence constraint between the presumed and the optimal planned trajectories implies that the coupling constraint (5)–(6) may not be satisfied. Indeed, these trajectories can be considerably different. On the other hand, the computation of each presumed trajectory of robots $R_p$ by vehicle $R_n$ increases the computation time and decreases the level of decentralization because of the knowledge of desired objectives $q_{p,fin}, \forall p \in \mathcal{C}_n(\tau_{k-1})$.
- A leader–follower receding horizon approach [19] where each robot sequentially computes its optimal planned trajectory. First, an order of resolution is determined to establish the level of priority for each robot. So, over any interval $[\tau_{k-1}, \tau_k)$, the robot which has the highest priority plans its optimal trajectory without taking the others into account. Once its trajectory is planned, another robot, having the second highest priority, generates its optimal planned trajectory by taking the already planned trajectories into account. This process is then repeated for every vehicle. The main advantage of this approach is its ease of implementation. However, because of the leader–follower architecture, leaders are single points of failure. Finally, when the velocity of the leader is maximal, the manoeuvre margin of the follower is limited.

In the following simulations, a Pentium IV (192 MB of RAM) micro-processor running at 2.4 GHz operating under Linux real time hosts the receding horizon motion planner written in C. The geometrical shape of each robot $R_n$ is included in a circle of radius $\rho_n = 0.2$ m. The linear and angular velocities are bounded, i.e. $|v_n| \leq 0.5$ m s$^{-1}$ and $|w_n| \leq 5$ rad s$^{-1}$. The map of the environment is constructed with the maximum range of 1.5 m. Each robot is assumed to track the planned trajectory well enough.

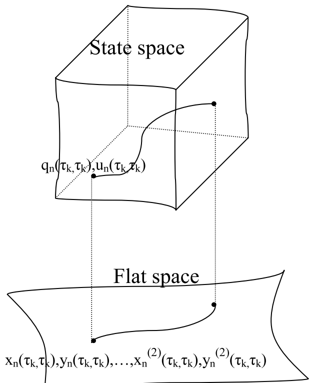



**Fig. 6.** Flatness and motion planning.

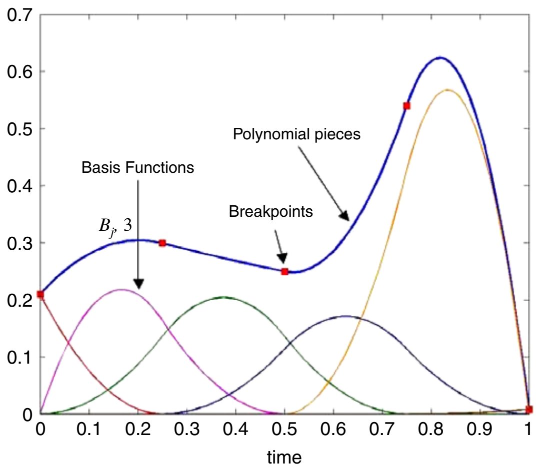



**Fig. 7.** Piecewise polynomial curve expressed in terms of a linear combination of three-order B-spline basis functions.

**Table 1**
Parameters of the receding horizon planners.

| | |
|---|---|
| $T_p$ | 2 s |
| $T_c$ | 0.5 s |
| $T_d$ | 2 s |
| $\xi$ | 0.25 |
| $n_{\text{knot}}$ | 5 |

### *4.2. Scenario 1: Crossing*

In this scenario, the group consists of two robots ($N = 2$) starting at $q_1(0) = [0, 0, 0]^T$ and $q_2(0) = [0, 5.1, 0]^T$, respectively, with velocities equal to zero. These robots must cross each other in order to reach their desired configuration $q_{1,fin} = [5, 5, 0]^T$ and $q_{2,fin} = [5, 0, 0]^T$, respectively, as quickly as possible. The parameters of the receding horizon planner are given in Table 1.

Here, we do not consider coupling constraint (6) on maximum separation distance (i.e. $d_{1,com} = d_{2,com} = +\infty$). Thus, only the collision avoidance (5) between robots is taken into account as a coupling constraint.

For robots $R_n$ ($n = 1, 2$), the goal is to determine, over each interval $[\tau_{k-1}, \tau_k)$, the optimal planned collision-free trajectory. One can note that, due to the symmetry of this problem, some properties of the studied algorithms can be highlighted.[4]

#### *4.2.1. Centralized and "weakly" decentralized approach [15]*

For the centralized strategy, a supervisor is used. At the initial time instance, the two robots send their initial configuration and their objective. Then, the supervisor solves, over any interval $[\tau_{k-1}, \tau_k)$, an optimization problem of dimension scale $2N(1 + n_{\text{knot}})$. The time needed to find the optimal solution when robots cross over is 172 ms. The simulation results are given in Fig. 8. The linear velocities, shown in Fig. 8(b), are less than 0.5 m/s. In Fig. 8(a)–(b), one can see that robots avoid collision by modifying their trajectory and their velocity. During the crossing, robots slow down and adapt their trajectory. Fig. 8(c) shows the evolution of the distance between robots. Since it is higher than 0.4 m, collision avoidance is guaranteed. The time to reach the desired objective is 15.8 s for robot $R_1$ and 16.2 s for robot $R_2$.

For the "weakly" decentralized approach [15], a supervisor is not needed. Each robot plans its own trajectory by solving an optimization problem of dimension scale $2(1 + n_{\text{knot}})$ as long as no collision may occur during the planning horizon. Then, when a collision may occur, each robot exchanges its last planned configuration and computes its trajectory and the one of its neighbor (optimization problem of size $2N(1 + n_{\text{knot}})$). Therefore, the results obtained are the same as the results using a centralized strategy for this scenario.

#### *4.2.2. Leader–follower approach [19]*

First, an order of resolution is determined (robot $R_1$, then robot $R_2$). Contrary to previous algorithms, leader $R_1$ does not try to avoid follower $R_2$. It plans its optimal trajectory by solving an optimization problem of dimension scale $2(1 + n_{\text{knot}})$ without taking the coupling constraints into account. Once its optimal trajectory is computed, it transmits it to robot $R_2$. Then, robot $R_2$ generates its optimal collision-free trajectory in order to avoid leader $R_1$. The time needed to solve the problem when robots cross over is small (about 40 ms). The simulation results are given in Fig. 9. The linear velocities, shown in Fig. 9(b), are less than 0.5 m/s. In Fig. 9(a)–(b), one can see that only robot $R_2$ modifies its trajectory and its velocity (i.e. deceleration) in order to avoid collision. Therefore, the time to reach the desired objective is 15.7 s for robot $R_1$ and 16.4 s for robot $R_2$. Fig. 9(c) shows the evolution of the distance between robots. Since it is higher than 0.4 m, collision avoidance is guaranteed.

#### *4.2.3. Proposed decentralized approach*

Contrary to the leader–follower algorithm, there is no order of resolution. The motion planning problem is solved in two steps. First, each robot computes a presumed trajectory. Then, they exchange their presumed trajectory if a collision may occur. Using the local information exchanged, they compute their optimal planned collision-free trajectory. The time needed to solve the problem when robots cross over is 94 ms. The simulation results are given in Fig. 10. The linear velocities, shown in Fig. 10(b), are less than 0.5 m/s. In Fig. 10(a)–(b), one can see that each robot modifies its trajectory and its velocity in order to avoid collision. The time to reach the desired objective is 16.0 s for robot $R_1$ and 16.3 s for robot $R_2$. Fig. 10(c) depicts the evolution of the distance between robots. Since it is higher than 0.4 m, collision avoidance is guaranteed.

**Remark 7.** Contrary to the "weakly" decentralized approach [15], constraint (21) is added in order to enforce the degree of correspondence between the optimal planned trajectory and the presumed trajectory known by the neighbors. Thus, collision avoidance between robots is guaranteed.

#### *4.2.4. Discussion*

The comparison in terms of the computation time, the amount of exchanged information and the traveling time for the team

[4] For comparison purposes, every optimal control problem is solved using the technique given in Section 3.3. The parameters of the receding horizon planner are the same for each algorithm studied.

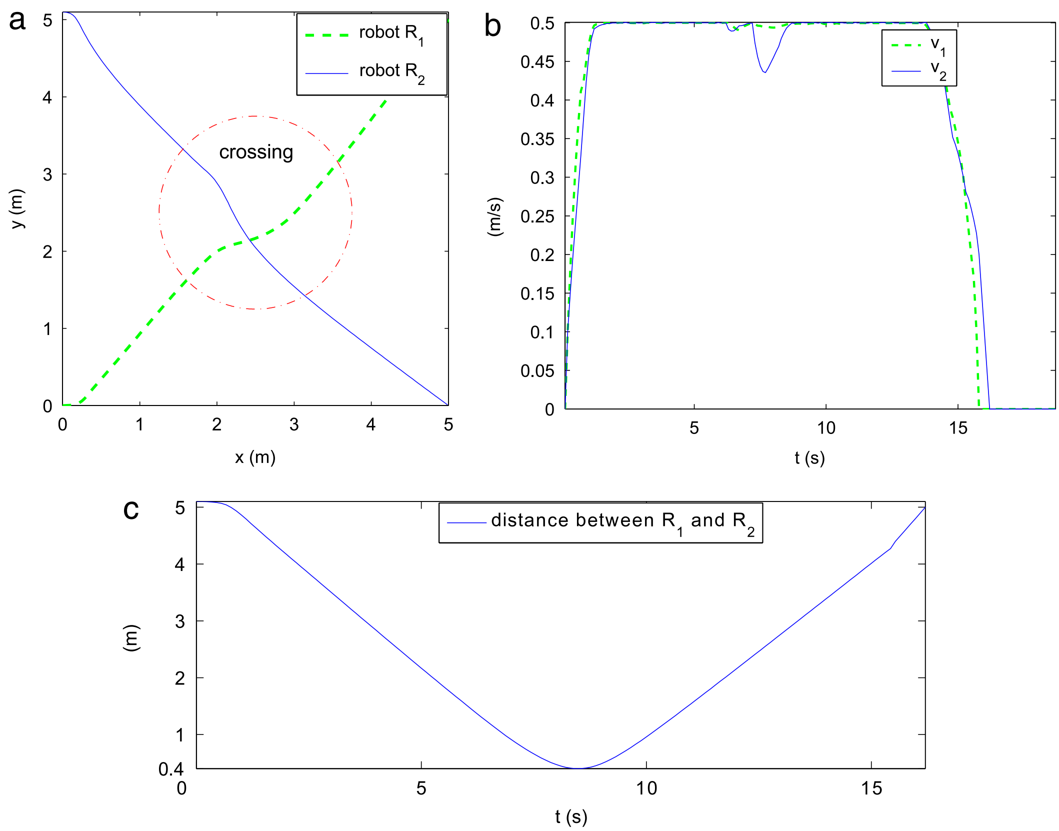



**Fig. 8.** Scenario 1: Centralized and weakly decentralized approaches.

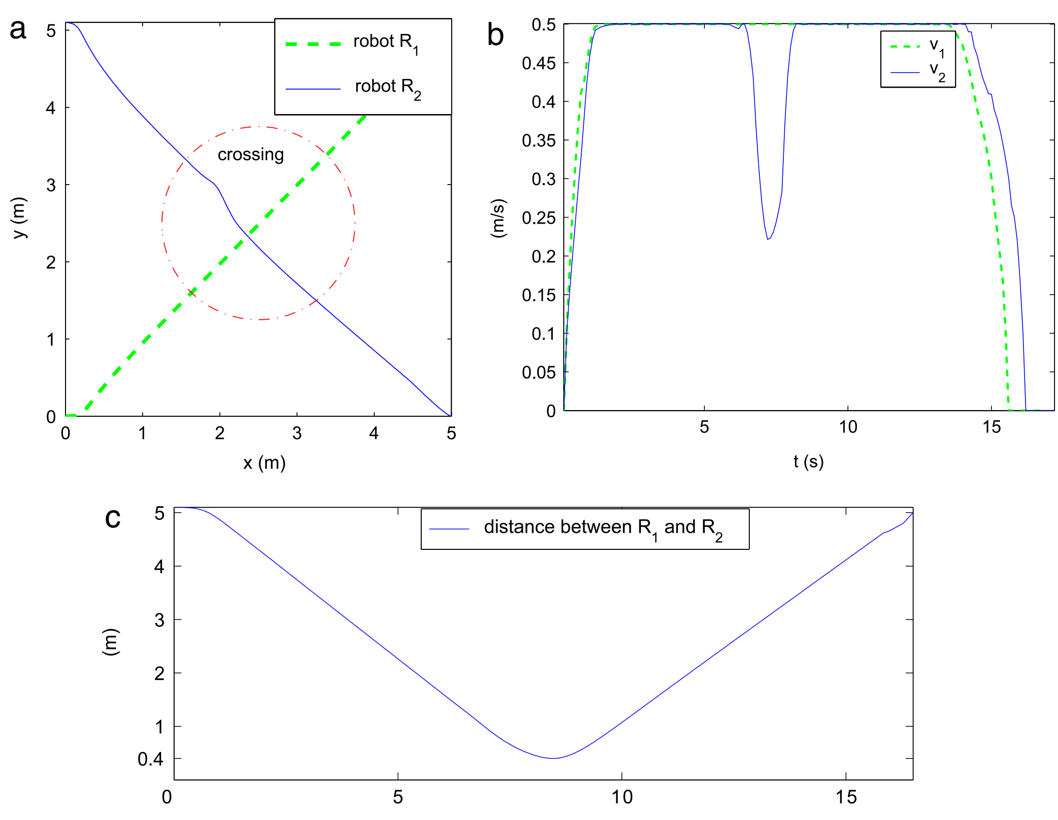



**Fig. 9.** Scenario 1: Leader–follower approach.

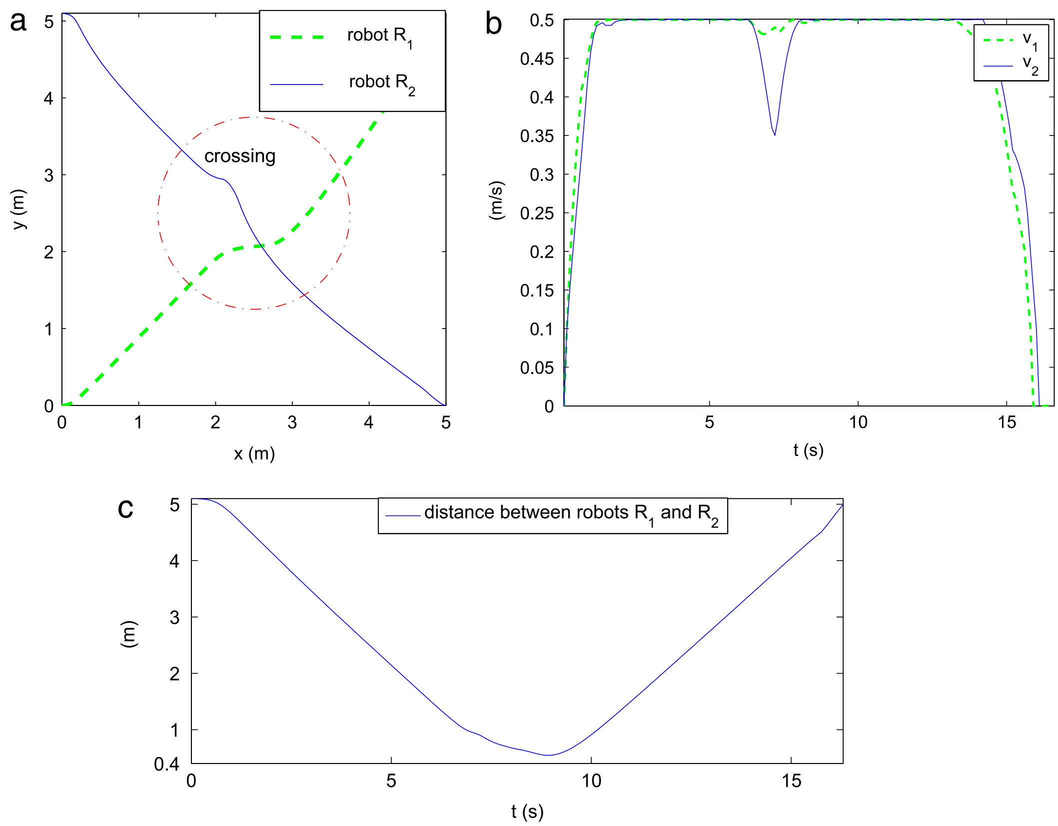



**Fig. 10.** Scenario 1: Proposed decentralized approach.

**Table 2**
Comparison of the proposed decentralized algorithm with other centralized and decentralized algorithms for scenario 1.

| Approach | Centralized | Leader–follower [19] | "Weakly" decentralized [15] | Proposed decentralized |
|---|---|---|---|---|
| Maximum computation time | 172 ms | 40 ms | 172 ms | 94 ms |
| Amount of exchanged information | 500 B | 50 B | 60 B | 100 B |
| Traveling time | 16.2 ms | 16.5 ms | 16.2 ms | 16.3 ms |

of two robots using the above algorithms is summarized in Table 2. For every algorithm, the maximum computation time (i.e. the time needed to solve the receding horizon problem when robots cross over) is less than the update period $T_c$. Hence, they are feasible on-line. Note that for the centralized and "weakly" decentralized algorithms, the computation time is about twice as big as the proposed strategy. For the centralized algorithm, the amount of exchanged information is more than five times as big as in the decentralized approaches. Compared to other existing receding horizon techniques, the proposed approach gives a more interesting compromise between optimality and computation time.

### *4.3. Scenario 2: Reconfiguration with collision avoidance*

In this scenario, the group consists of five robots ($N = 5$) starting at the initial configurations given in Table 3, with velocities equal to zero. These robots must reach the desired configuration given in Table 4 as quickly as possible. One can note that the geometrical shape of the group will be reconfigured (from "linear" to "triangular" shape). All along the way, they need to maintain the connectivity, i.e. there are communication links between the pairs $(R_1, R_2)$, $(R_2, R_4)$, $(R_1, R_3)$ and $(R_3, R_5)$ (see the communication graph given in Fig. 11). That is why the robots must not exceed the maximum distance $d_{com} = 2.5$ m from their neighbors. The five robots make decisions in order to avoid collisions with static obstacles and with other robots, and to maintain the communication links.

**Table 3**
Initial configurations ("linear" shape).

| Robot | $x$-position (m) | $y$-position (m) | $\theta$-position (rad) |
|---|---|---|---|
| $R_1$ | 0 | 0 | 0 |
| $R_2$ | 0 | 2 | 0 |
| $R_3$ | 0 | −2 | 0 |
| $R_4$ | 0 | 4 | 0 |
| $R_5$ | 0 | −4 | 0 |

**Table 4**
Desired configurations ("triangular" shape).

| Robot | $x$-position (m) | $y$-position (m) | $\theta$-position (rad) |
|---|---|---|---|
| $R_1$ | 15 | 0 | 0 |
| $R_2$ | 13.5 | −1.5 | 0 |
| $R_3$ | 13.5 | 1.5 | 0 |
| $R_4$ | 12 | −3 | 0 |
| $R_5$ | 12 | 3 | 0 |

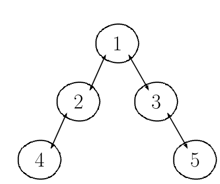



**Fig. 11.** Communication graph.

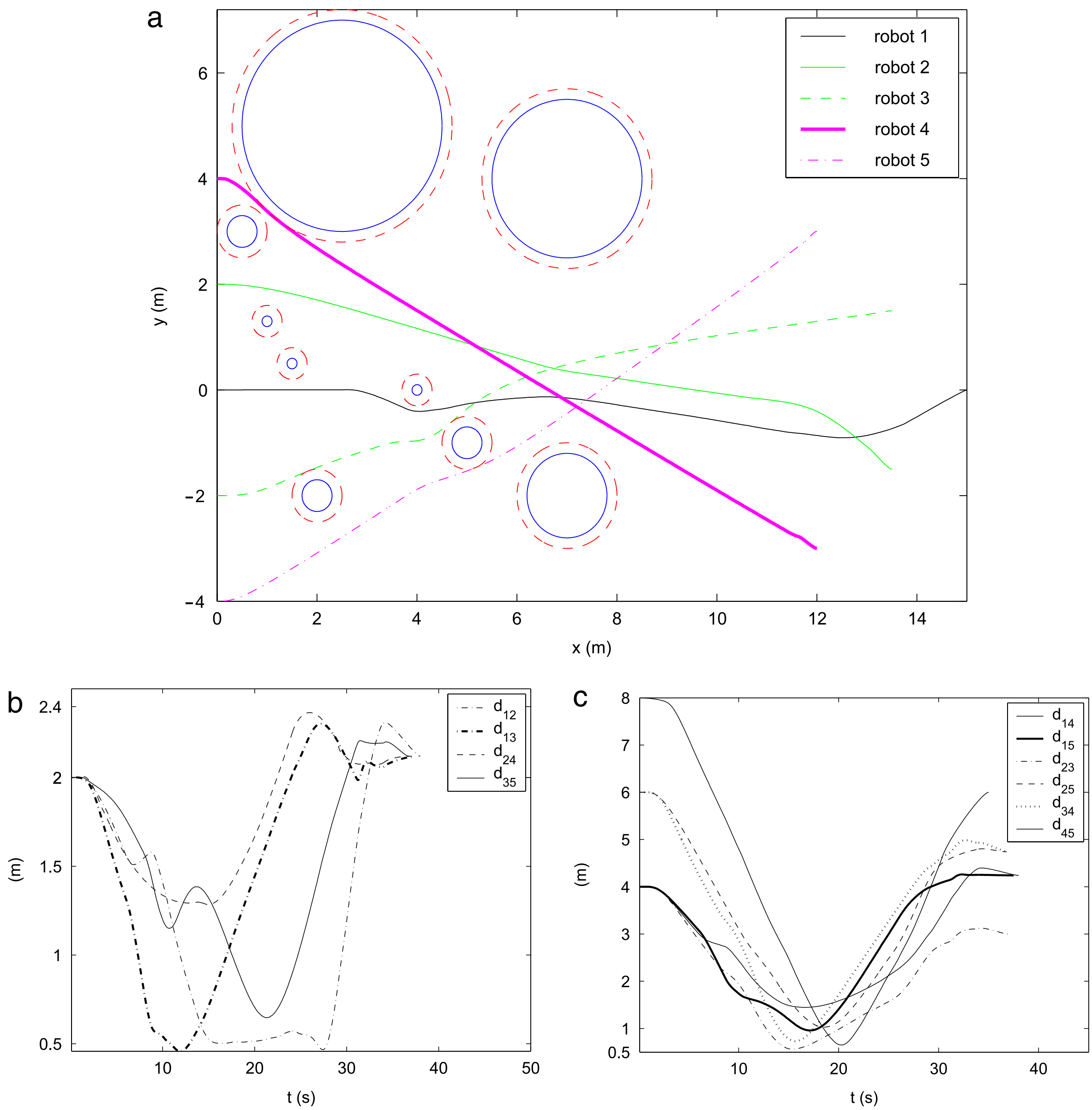



**Fig. 12.** Scenario 2: Proposed decentralized approach.

**Table 5**
Parameters of the receding horizon planners.

| | |
|---|---|
| $T_p$ | 2 s |
| $T_c$ | 0.5 s |
| $T_d$ | 2.5 s |
| $\xi$ | 0.2 5 |
| $n_{\text{knot}}$ | 5 |

The parameters of the receding horizon planner are given in Table 5. One can note that, due to the number of potential conflicts during crossings, the advantages and limitations of the algorithms studied can be highlighted.

The simulation results are given in Fig. 12. One can see that each robot modifies its trajectory and its velocity in order to satisfy all the constraints (i.e. nonholonomic constraints, limitations on velocities, obstacle avoidance,[5] collision avoidance between robots and constraints on communication links). The coordination is done by mutual adjustments (i.e. there is no leader). The time to reach the desired objective is 36.5 s for the group. Fig. 12(b) shows the evolution of the distances between robots which must maintain a communication link. One can see that these distances are higher than 0.4 m and lower than 2.5 m. The other relative distances are given in Fig. 12(c), showing the collision avoidance between robots.

**Fig. 13.** The Pekee nonholonomic mobile robots.

[5] In order to take the size of robots into account, the radius of obstacles is increased by 0.2 m (dotted lines around obstacles).

**Table 6**
Comparison of the proposed decentralized algorithm with other centralized and decentralized algorithms for scenario 2.

| Approach | Centralized | Leader–follower [19] | "Weakly" decentralized [15] | Proposed decentralized |
|---|---|---|---|---|
| Maximum computation time | 2050 ms | 313 ms | 703 ms | 121 ms |
| Amount of exchanged information | 5000 B | 1250 B | 1200 B | 2650 B |
| Level of decentralization | No | + | − | + |
| Feasibility | No | Yes | No | Yes |
| Traveling time | 35 s | 39 s | 36 s | 36.5 s |

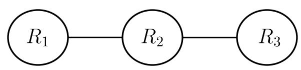



**Fig. 14.** Communication graph for the experiment.

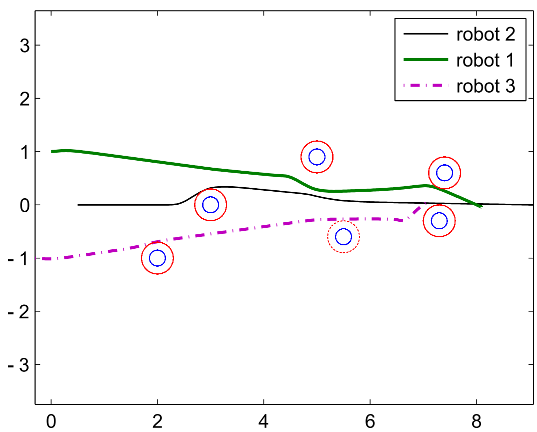



**Fig. 15.** Resulting robot trajectories.

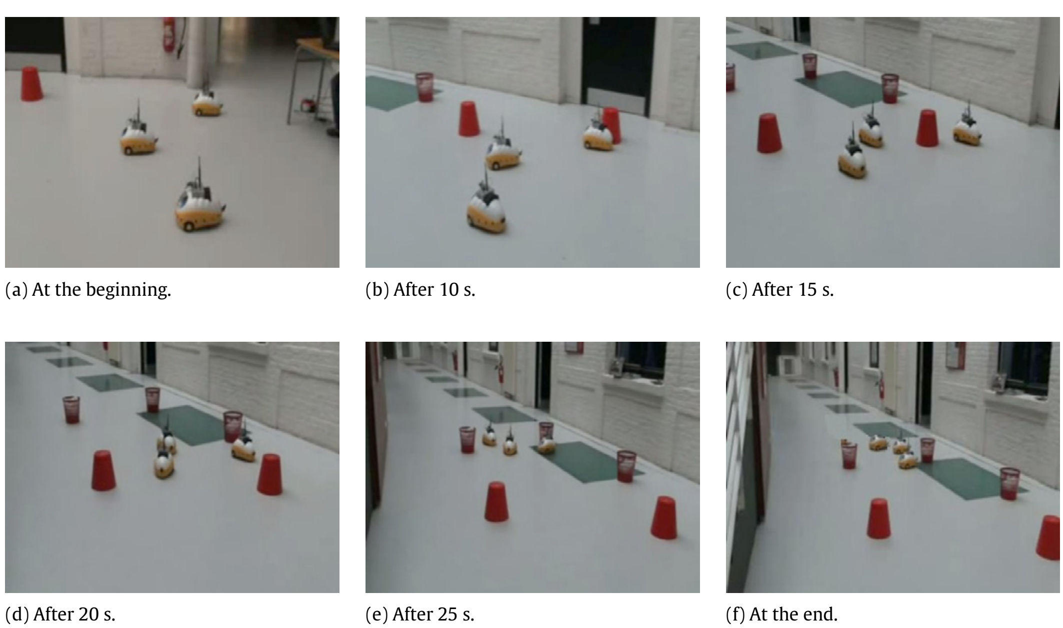
Finally, in order to show the advantages of the proposed decentralized algorithm, the same five-vehicle scenario is tested using other receding horizon algorithms. The comparison in terms of maximum computation time for the design of the optimal trajectory during one planning horizon, the amount of exchanged information, the traveling time, the level of decentralization and the feasibility using the studied algorithms is summarized in Table 6.

The centralized algorithm provides the best results in terms of traveling time. However, since the maximum computation time for the design of the optimal trajectory during one planning horizon is higher than update horizon $T_c$, this algorithm cannot be applied on-line, i.e. it is infeasible. The same conclusion arises from the "weakly" decentralized algorithm due to the high dimension scale of optimization problems during conflicts. Here, only, the proposed decentralized and the leader–follower algorithms can be applied in real time. However, using the leader–follower approach, the optimality is damaged because of the lack of information from the follower to the leader. This lack of cooperation implies a more difficult motion planning for robots with low priority. The proposed decentralized algorithm circumvents this problem while increasing the amount of exchanged information between robots. This strategy remains feasible and the amount of exchanged information is twice less than the centralized approach. Hence, one can conclude that the proposed decentralized algorithm provides a good compromise between optimality and computation time for this scenario compared to other existing receding horizon algorithm.

## 5. Experimental results

The experiments are performed with a team of three nonholonomic mobile robots which are two-wheel differentially driven robots, as shown in Fig. 13. The geometrical shape of each robot $R_n$ is included in a circle of radius $\rho_n = 0.2$ m. Robot Pekee is equipped with 15 infra-red telemeters sensors, two encoders, a WiFi wireless cartridge and a miniature color vision camera, C-Cam8.

A limitation of the embedded vision camera is that its resolution quickly decreases with the distance of the objects. Indeed, the range of this sensor is about 1.5 m. The localization and mapping method reported in [36] is applied. In order to track the optimal planned trajectory, each robot uses the robust closed-loop controller described in [37].



(a) At the beginning. (b) After 10 s. (c) After 15 s.

(d) After 20 s. (e) After 25 s. (f) At the end.

**Fig. 16.** Three Pekee mobile robots flocking from initial configurations to a desired region, avoiding collisions and maintaining communication constraints.

We also consider communication constraints given by

$$\begin{cases} d(q_1, q_2) \leq d_{com}, \\ d(q_2, q_3) \leq d_{com}, \end{cases} \tag{25}$$

where $d_{com} = 2.5$ m is the broadcasting range of each robot. The associated communication graph is depicted in Fig. 14.

The decentralized receding horizon motion planner whose parameters are the same as in the simulation, i.e. given in Table 5, is tested in the experiment. At the beginning, the geometrical shape of the group is triangular. The task is to drive these robots to form a desired triangular shape while avoiding collisions and maintaining the communication constraints. Due to the existence of obstacles, robots must pass through narrow ways and constraint each other in the team.

One can see in Fig. 15 the resulting robot trajectories. Fig. 16 shows six snapshots of our experiment. During the motion, each robot computes its optimal planned trajectory using local exchanged information. Fig. 16(a) depicts the three robots at the beginning. In Fig. 16(a)–(f), they approach their goal positions to form the desired geometrical shape while avoiding collision and maintaining communication links. This experiment demonstrates that the proposed receding horizon planner manages to accomplish the desired objectives.

## 6. Conclusion

A decentralized algorithm, based on a receding horizon approach, is proposed in order to solve the motion planning problem for cooperative unicycle-type mobile robots subject to constraints imposed by sensors and the communication network. The main idea is to divide into two steps the generation of the optimal trajectory which satisfies the set of all the constraints for each robot. In the first step, a presumed trajectory is computed without taking the coupling constraints into account. In the second step, each robot elaborates its optimal planned trajectory by taking the intentions of neighboring vehicles into account. Examples of crossing and reconfiguration highlight the advantages of the proposed strategy, especially in terms of computation time, traveling time and feasibility, compared to other existing receding horizon approaches. Experimental results show the effectiveness of the proposed decentralized algorithm.

## References


[1] A. Broggi, M. Bertozzi, A. Fascioli, C. Bianco, A. Piazzi, Visual perception of obstacles and vehicles for platooning, IEEE Transactions on Intelligent Transportation Systems 1 (2000) 164–176.
[2] N. Miyata, J. Ota, T. Arai, H. Asama, Cooperative Transport by Multiple Mobile Robots in Unknown Static Environments Associated With Real-Time Task Assignment, IEEE Transactions on Robotics and Automation 18 (2002) 769–780.
[3] T. Balch, R. Arkin, Behavior-based formation control for multirobot teams, IEEE Transactions on Robotics and Automation 14 (1998) 926–939.
[4] M. Defoort, T. Floquet, A. Kokosy, W. Perruquetti, Sliding mode formation control for cooperative autonomous mobile robots, IEEE Transactions on Industrial Electronics 55 (2008) 3944–3953.
[5] G. Inalhan, D. Stipanovic, C. Tomlin, Decentralized optimization, with application to multiple aircraft coordination, in: Proc. IEEE Conf. Decision and Control, 2001.
[6] J. Kuchar, L. Yang, A review of conflict detection and resolution modeling methods, IEEE Transactions on Intelligent Transportation Systems 1 (2000) 179–189.
[7] O. Purwin, R. D'Andrea, J. Lee, Theory and implementation of path planning by negotiation for decentralized agents, Robotic and Autonomous Systems 56 (2008) 422–436.
[8] K. Do, Output-feedback formation tracking control of unicycle-type mobile robots with limited sensing ranges, Robotic and Autonomous Systems 57 (2009) 34–47.
[9] M. Defoort, J. Palos, A. Kokosy, T. Floquet, W. Perruquetti, Performance-based reactive navigation for nonholonomic mobile robots, Robotica 27 (2009) 281–290.
[10] I. Harmati, K. Skrzypczyk, Robot team coordination for target tracking using fuzzy logic controller in game theoretic framework, Robotic and Autonomous Systems 57 (2009) 75–86.
[11] E. Frazzoli, Z. Mao, J. Oh, E. Feron, Resolution of conflicts involving many aircraft via semidefinite programming, Journal of Guidance, Control, and Dynamics 24 (2001) 79–86.
[12] A. Raghunathan, V. Gopal, D. Subramanian, L.T. Biegler, T. Samad, Dynamic optimization strategies for 3D conflict resolution of multiple aircraft, Journal of Guidance, Control, and Dynamics 27 (2004) 586–594.
[13] F. Borelli, D. Subramanian, A. Raghunathan, L. Biegler, T. Samad, MILP and NLP techniques for centralized trajectory planning of multiple unmanned air vehicles, in: Proc. Amer. Control Conf., 2006.
[14] W. Dunbar, R. Murray, Distributed receding for multi-vehicle formation stabilization, Automatica 42 (2006) 549–558.
[15] T. Keviczky, F. Borelli, G. Balas, Decentralized receding horizon control for large scale dynamically decoupled systems, Automatica 42 (2006) 2105–2115.
[16] T. Keviczky, F. Borelli, K. Fregene, D. Godbole, G. Balas, Decentralized receding horizon control and coordination of autonomous vehicle formations, IEEE Transactions on Control Systems Technology 16 (2008) 19–32.
[17] T. Schouwenaars, J. How, E. Feron, Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees, in: Proc. AIAA Guidance, Navigation and Control Conf., 2004.
[18] A. Richards, J. How, Decentralized model predictive control of cooperating UAVs, in: Proc. IEEE Conf. Decision and Control, 2004.
[19] Y. Kuwata, A. Richards, T. Schouwenaars, J. How, Distributed robust receding horizon control for multivehicle guidance, IEEE Transactions on Control Systems Technology 15 (2007) 627–641.
[20] M. Bennewitz, W. Burgard, S. Thrun, Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots, Robotic and Autonomous Systems 41 (2002) 89–99.
[21] S. LaValle, S. Hutchinson, Optimal motion planning for multiple robot having independent goals, IEEE Transactions on Robotics and Automation 14 (1998) 912–925.
[22] G. Sanchez, J. Latombe, On delaying collision checking in PRM planner to compare centralized and decoupled planning for multi-robot systems, International Journal of Robotics Research 21 (2002) 5–26.
[23] D. Dimarogonas, K. Kyriakopoulos, Decentralized navigation functions for multiple robotic agents with limited sensing capabilities, Journal of Intelligent and Robotic Systems 48 (2007) 277–288.
[24] G. Pereira, A. Das, V. Kumar, M. Campos, Decentralized motion planning for multiple robots subject to sensing and communication constraints, in: Proc. Workshop Multi-Robot Systems, 2003.
[25] G. Pereira, V. Kumar, M. Campos, Closed loop motion planning of cooperating mobile robots using graph connectivity, Robotic and Autonomous Systems 56 (2008) 373–384.
[26] A. Masoud, Decentralized, self-organizing, potential field-based control for individually-motivated, mobile agents in a cluttered environment: A vector-harmonic potential field approach, IEEE Transactions on Systems, Man & Cybernetics, Part A (Systems & Humans) 37 (2007) 372–390.
[27] C. Canudas de Wit, O. Sordalen, Exponential stabilization of mobile robots with nonholonomic constraints, IEEE Transactions on Automatic Control 37 (1992) 1791–1797.
[28] K. Lerman, C. Jones, A. Galstyan, M. Mataric, Analysis of dynamic task allocation in multi-robot systems, International Journal of Robotics Research 25 (2006) 225–241.
[29] A. Das, J. Spletzer, V. Kumar, C. Taylor, Ad hoc networks for localization and control, in: Proc. IEEE Conf. Decision and Control, 2002.
[30] F. Cuzzolaa, J. Geromel, M. Mrari, An improved approach for constrained robust model predictive control, Automatica 38 (2002) 1183–1189.
[31] M. Defoort, T. Floquet, A. Kokosy, W. Perruquetti, A novel higher order sliding mode control scheme, Systems and Control Letters 58 (2009) 102–108.
[32] M. Fliess, J. Levine, P. Martin, P. Rouchon, Flatness and defect of nonlinear systems: Introductory theory and examples, International Journal of Control 61 (1995) 1327–1361.
[33] C. de Boor, A Practical Guide to Splines, Springer-Verlag, 1978.
[34] C. Lawrence, A. Tits, A computationally efficient feasible sequential quadratic programming algorithm, SIAM Journal on Optimization 11 (2001) 1092–1118.
[35] M. Milam, Real-time optimal trajectory generation for constrained dynamical systems, Ph.D. Dissertation, California Inst. of Technology, 2003.
[36] M. Defoort, J. Palos, A. Kokosy, T. Floquet, W. Perruquetti, D. Boulinguez, Experimental motion planning and control for an autonomous nonholonomic mobile robot, in: Proc. IEEE Conf. Robotics and Automation, 2007.
[37] M. Defoort, T. Floquet, A. Kokosy, W. Perruquetti, Integral sliding mode control for trajectory tracking of a unicycle type mobile robot, Integrated Computer-Aided Engineering 13 (2006) 277–288.


**Michael Defoort** was born in France in 1981. He received his M.S. degree in automatic control from Institut Supérieur d'Electronique et du Numérique, Lille, France, in 2004, and his Ph.D. degree from Ecole Centrale de Lille, Lille, France, in 2007. From 2007 to 2008, he was a Visiting Researcher in the Department of System Design Engineering, Keio University, Yokohama, Japan. From 2008 to 2009, he was a postdoctoral researcher in the Department of Computer Science and Automatic Systems, Ecole des Mines de Douai, Douai, France. He is currently an Assistant Professor in the Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines LAMIH (UMR-CNRS 8530),

Valenciennes University, 59313 Valenciennes, France. His research interests include nonholonomic systems, nonlinear control, motion planning and formation control.

**Annemarie Kokosy** was born in Romania in 1970. She received her Diploma in automatic control engineering from the University ''Politechnica'' Timisoara, Romania, in 1994, her M.S. degree from the Ecole Nationale d'Ingénieurs de Belfort, Belfort, France, in 1995, and her Ph.D. degree from the University Franche-Comté, Besançon, France, in 1999. From 1998 to 1999, she was an Assistant Professor with the Department of Automatic Control, University Franche-Comté. From 1999 to 2000, she was an Assistant Professor with the Department of Automatic Control, University of Lille, Lille, France. Since May 2000, she has been with the Department of Signals and Systems, Institut Supérieur d'Electronique et du Numérique, Lille, France, where she is currently an Assistant Professor. Since 2003, she has been the Head of the Department of Signals and Systems. She is also with the Laboratoire d'Automatique, Génie Informatique et Signal, Unité Mixte de Recherche 8146, Centre National de la Recherche Scientifique, Ecole Centrale de Lille. Her research interests include robotics and analysis and control for nonlinear systems.

**Thierry Floquet** received his M.S. and Ph.D. degrees in automatic control from the University of Lille, Lille, France, in 1997 and 2000, respectively. Since 2001, he has been with the Laboratoire d'Automatique, Génie Informatique et Signal, UMR CNRS 8146, Ecole Centrale de Lille, Villeneuve-d'Ascq, France, where he is Chargé de Recherche of the Centre National de la Recherche Scientifique. His research interests include observer theory and nonlinear control with applications to electrical engineering and robotics.

**Wilfrid Perruquetti** was born in Saint Gilles, France, in 1968. He received his M.Sc. degree in automatic control from the ''Institut Industriel du Nord'' (French ''Grande Ecole''), Lille, France, in 1991. He obtained his Ph.D. degree in automatic control in 1994. He joined the Ecole Centrale de Lille, Lille, as an Assistant Professor in 1995, where he has been a Full Professor since 2003 and is working in the Laboratoire d'Automatique, Génie Informatique et Signal, UMR CNRS 8146 and within the project ALIEN of INRIA-Lille Nord Europe. He has published about 128 book chapters, journal, and conference papers and is the Co-editor of two books with Jean-Pierre Barbot: Sliding Mode Control in Engineering (Marcel Dekker) and Chaos in Automatic Control (CRC-Taylor and Francis). His research interests include stability analysis, (finite time) stabilization (in particular, stabilization), sliding-mode control of nonlinear and time-delay systems, and observation and identification of system parameters. His main fields of applications are in robotics (mobile robots and robotic manipulators) and electrical actuators (DC, induction, and stepper motors). Prof. Perruquetti has been a member of two International Federation of Automatic Control Technical Committees (2.3 ''Nonlinear Control Systems'' and 2.5 ''Robust Control'') since 2002, and he is a Senior Member of the Société de l'Electricité, de l'Electronique et des Technologies de l'Information et de la Communication. In 2007, he was appointed as a representative of the Direction Générale de la Recherche et de l'Innovation by the French Ministry of Education and Research. He was the Program Director of several programs and was involved in several International Program Committees. He is currently the Editor-in-Chief of e-revue Sciences et Technologie de l'Automatique.

**Jorge Palos** received his Ph.D. degree in automatic control from the University of Lille, Lille, France, in 2004. His Ph.D work focused on image processing in medical applications. He is currently an Assistant Professor with the department of Systems and Telecommunications at Institut Supérieur d'Electronique et du Numérique, Lille, France. His research interests include application of image processing, stereo vision and vehicle guidance.