



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

Michael Defoort, Thierry Floquet, Annemarie Kökösy, Wilfrid Perruquetti

► To cite this version:

Michael Defoort, Thierry Floquet, Annemarie Kökösy, Wilfrid Perruquetti. Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: A distributed receding horizon approach. *Robotics and Autonomous Systems*, 2009, 57 (11), pp.1094-1106. 10.1016/j.robot.2009.07.004 . hal-00519825

HAL Id: hal-00519825

<https://hal.science/hal-00519825>

Submitted on 21 Sep 2010

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

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

Decentralized motion planning for cooperative nonholonomic mobile robots^{*}

Michael Defoort^{*,a} Annemarie Kokosy^b Thierry Floquet^c
Wilfrid Perruquetti^c

^a*Department of System Design Engineering, Keio University, Yokohama, Japan*

^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,
59651 Villeneuve d'Ascq, France*

Abstract

This paper presents a decentralized motion planner for a formation of autonomous mobile robots evolving in an unknown environment with obstacles. The motion planning scheme consists of decentralized receding horizon planners that reside on each vehicle to achieve coordination among formation 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 assumed trajectory designed without taking the coupling constraints into account. Finally, 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.

Key words: Multi-vehicle formation, Decentralized intelligence, Receding horizon planning, Nonholonomic mobile robot.

1 Introduction

The research effort in multi-agent system relies on the fact that multiple agents have the possibility to solve problems more efficiently than a single agent. Co-

^{*} This work was partially supported by the JSPS Postdoctoral Fellowship and the FEDER (European Funds of Regional Development) under the ARCIR Robocoop and the AUTORIS-TAT T31 projects.

^{*} Corresponding author. E-mail: defoort@sum.sd.keio.ac.jp

operative control of multiple vehicles gives rise to significant theoretical challenges and has various engineering applications including cooperative transportation of a large payload [18], formation flying of unmanned aerial vehicles [10], multi-agent gaming such as robot soccer or robot rescue [1,14,19].

In this paper, the problem of interest is the navigation of a formation of autonomous 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, some communication links between some vehicles must be maintained during the movement. Since the available power onboard the vehicle is limited, the distance between two vehicles which may exchange information will naturally be constrained. Besides maintaining the communication, the feasibility of the trajectories implies the 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 environment is usually partially known and further explored in real time. Therefore, the trajectories have to be computed gradually over time while the mission unfolds. It can be accomplished using an online receding horizon planner [16], in which partial trajectories from an initial state toward the goal are computed by solving an optimal control problem over a limited horizon.

Two strategies for motion planning in multi-agent systems are the centralized and decentralized (distributed) approaches. Although the centralized one has been used in different studies (see [7] for instance), its computation time which scales exponentially with the number of vehicles, its communication requirement and its lack of security make it prohibitive. To overcome these limitations, one can use a distributed strategy which results in a formation behavior similar to what is obtained with a centralized approach.

Recently, some decentralized receding horizon planners have been proposed in [8,11–13]. In [8], a solution is provided for unconstrained subsystems (decoupled input constraints only). Therefore, the coupling constraints between robots cannot be taken into account. In [11,12], a distributed planner is formulated where each robot optimizes locally for itself as well as for every neighbor at each update, resulting in the increase in the computing time and the decrease in the decentralization. Furthermore, in order to ensure collision avoidance, some emergency strategies must be defined. In [13], the decentralized

scheme is based on a leader-follower architecture. Indeed, the vehicles update their trajectory sequentially and the feasibility is guaranteed in spite of the presence of coupling constraints between subsystems. One advantage of the leader-follower 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 based on potential fields are given in [5,6] but they do not enable to satisfy the nonholonomic constraint imposed by the rolling wheels.

In this paper, a distributed implementation, without assigning any leader, of receding horizon planning is presented. Each robot optimizes only for its own trajectory at each update and exchanges information with neighboring subsystems. It requires the exchange of the assumed trajectory designed without taking the coupling constraints into account between neighboring subsystems prior to each update. Providing that the robot does not deviate too far from the assumed trajectory, a conflict-free trajectory satisfying all the constraints can be found. Contrary to [5,6,8,11,12], 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. In Section 3, our decentralized algorithm which only uses local information is presented. Finally, 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.

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 centre of mass is at the geometric centre C_n of the body. It has two driving wheels fixed to the axis which passes through C_n and one passive centred 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 motion of the passive wheel can be ignored in the dynamics of the mobile robot. The centre 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:

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

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

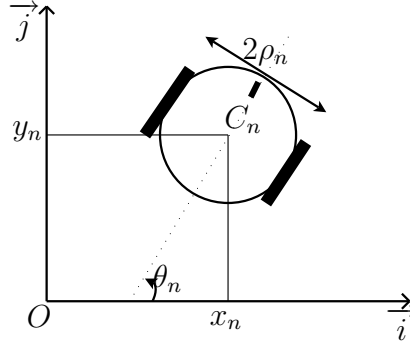


Fig. 1. Unicycle-type mobile robot

In this paper, the kinematics of wheeled-mobile robot is shown under the nonholonomic constraints (see [2] for details). The pure rolling and nonslipping nonholonomic conditions are described by:

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

The kinematic equations can be written as follows:

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

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

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

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

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

2.2 Problem setup

The following assumptions are made in this study:

- (i) robots have on-board sensors which can detect surrounding objects and vehicles within a range with a small margin of error,
- (ii) robots can reliably communicate with each other,
- (iii) the broadcasting range is limited,
- (iiii) 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).

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

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

where $L_n(q_n, u_n, q_{n,fin}) : \mathbb{R}^3 \times \mathbb{R}^2 \times \mathbb{R}^3 \mapsto \mathbb{R}$ 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 $u_n = 0$), $q = [q_1^T, \dots, q_N^T]^T$ and $u = [u_1^T, \dots, u_N^T]^T$ are the concatenated trajectory and control input. To solve the motion planning problem, one can make the following choices without loss of any generality¹:

- The geometric shape of robot R_n is represented by a 2-D circle of centre $C_n = (x_n, y_n)$ and of radius ρ_n . Its motion is controlled but nonholonomic and is represented by the velocity vector $u_n(t)$. The range of its sensors is also described by a circle centered at C_n .
- The i^{th} ($i = 1, \dots$) obstacle in the environment will be represented by a circle centered at point $O_i = (X_i, Y_i)$ and of radius r_i , denoted by $\mathcal{B}_i(O_i, r_i)$.

Definition 1 $\forall t_k \in [t_{ini}, t_{fin}]$, $\forall R_n$ ($n \in \{1, \dots, N\}$), 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 Fig. 2-3). 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:

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

Here, a cooperative control problem is considered where some constraints couple the dynamic behavior of the robots. Indeed, the trajectory feasibility implies that the topology of the wireless network is kept at any time, i.e some

¹ It is trivial to allow the envelope of either the robot or 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.

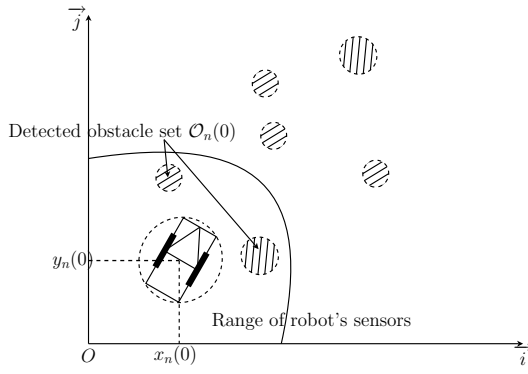


Fig. 2. Detected obstacle set at time $t_k = 0s$.

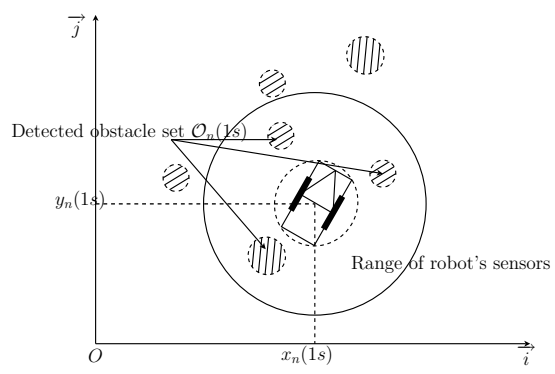


Fig. 3. Detected obstacle set at time $t_k = 1s$.

communication links between some vehicles must be maintained during the movement. These coupling constraints can be expressed by the communication graph defined as follows.

Definition 2 A communication graph $(\mathcal{R}, \mathcal{E}, \mathcal{S})$ is a labeled graph consisting of:

- a set $\mathcal{R} = \{R_1, \dots, R_N\}$ of N nodes representing the robots.
- a set of edges $\mathcal{E} \subset \mathcal{R} \times \mathcal{R}$ encoding the communication links. Pair $R_n - R_p$ belongs to \mathcal{E} if there is a communication link between robots R_n and R_p .
- a set of edge constraints. Since the available power onboard the vehicle is limited, the distance between two vehicles which may exchange information is constrained. Let us denote the broadcasting range of each R_n as $d_{n,com}$ (> 0). For each pair $(R_n, R_p) \in \mathcal{E}$, the distance $d(q_n(t), q_p(t)) = \sqrt{(x_n(t) - x_p(t))^2 + (y_n(t) - y_p(t))^2}$ has to be smaller than the shortest range of the two robots, i.e:

$$d(q_n(t), q_p(t)) \leq \min(d_{n,com}, d_{p,com}). \quad (5)$$

Lastly, the robots should 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 $(R_n, R_p) \in \mathcal{R} \times \mathcal{R}$, $n \neq p$,

$$d(q_n(t), q_p(t)) \geq \rho_n + \rho_p. \quad (6)$$

The objective is to compute the optimal trajectory, for each vehicle, from $q_n(t_{ini})$ to $q_{n,fin}$ which satisfies requirements (1), (2) and (4)-(6) outlined above, in a decentralized cooperative way and according to the performance criteria (3). Depending on the distance that robots have to travel, the computation of complete trajectories from start until finish may be computationally too expensive. Moreover, the environment is partially known and further explored in real time. Therefore, the trajectory have 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 the next section, we propose a way to decentralize the receding horizon planning problem in order to reduce the computational complexity and communication requirements.

3 Decentralized motion planning scheme

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 receding horizon planning problem. At each update, denoted τ_k ($k \in \mathbb{N}$),

$$\tau_k = t_{ini} + kT_c, \quad (7)$$

each robot computes an optimal trajectory satisfying constraints (1), (2) and (4)-(6) using local information. One idea is to only include, for each receding horizon planner, the robots that could have direct conflicts (i.e. may produce collision or may lose the communication). It enables a decentralized algorithm based on local information.

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:

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

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

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

Remark 1 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 $[\tau_k, \tau_{k+1} + T_p]$.

In order to avoid a leader-follower approach and to strongly decentralize the algorithm, it is proposed that each vehicle R_n only plans its own trajectory using local information.

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 trajectories for robots R_p belonging to the conflict set in order to plan its optimal collision-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 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 optimal control problem into two steps. First, each robot R_n computes its presumed trajectory, denoted $\hat{q}_n(t, \tau_k)$, over the prediction horizon (i.e. $t \in [\tau_k, \tau_k + T_d]$) and its corresponding control input, denoted $\hat{u}_n(t, \tau_k)$. This trajectory is obtained without taking the coupling constraint (5)-(6) into account. Then, each vehicle transmits its presumed trajectory and control inputs to all the robots belonging to its conflict set. Therefore, by design, the presumed trajectory is the same in every distributed optimal control problem in which it occurs, i.e. $\forall R_p \in \mathcal{C}_n(\tau_k)$, the same trajectory is presumed for R_n . Then, using only local information, each robot computes its optimal trajectory, denoted $q_n^*(t, \tau_k)$, which satisfies constraints (1), (2) and (4)-(6) and the associated control inputs, denoted $u_n^*(t, \tau_k)$, over the planning horizon (i.e. $t \in [\tau_k, \tau_k + T_p]$).

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

Problem $\hat{P}_n(\tau_k)$:

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

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

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

Remark 2 During the initialization step, that is to say before 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 $\hat{q}_n(t, \tau_k)$ 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 $P_n^*(\tau_k)$:

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

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

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

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

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

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

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

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

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

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

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

where $\xi \in \mathbb{R}^+$ is a constant. Here, it is assumed that the optimal planned trajectory $q_n^*(t, \tau_k)$ is constrained to be at most at a distance ξ from the presumed trajectory $\hat{q}_n(t, \tau_k)$ (see eq. (21)). This constraint enforces the degree

of correspondence between the optimal trajectory and the presumed trajectory known by the neighbors. Therefore, in order to guarantee the collision avoidance between vehicles and the preservation of the communication links, deformation ξ due to the mismatch between the presumed and the optimal planned trajectories is added (see equations (19)-(20)).

Remark 3 *From the optimal planned trajectory and control inputs associated to the planning horizon T_p (i.e. $q_n^*(t, \tau_k)$ and $u_n^*(t, \tau_k)$), only the part which corresponds to the update horizon T_c is stored (see Fig. 4).*

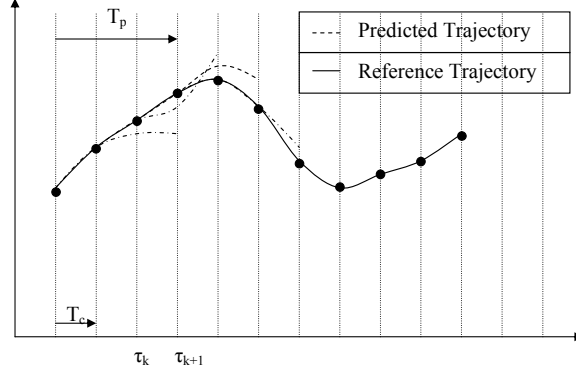


Fig. 4. Planning and update horizons

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 robots 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 neighbourhood of their goal $q_{n,fin}$. As such, new information can be taken into account in the next iteration.

Remark 4 *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, 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 [3]). 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. 6).*

The discussed claim for robustness in trajectory tracking can be effectively accomplished by sliding mode control. Robustness properties against various kinds of uncertainties such as parameter perturbations, external disturbances and measurement errors can be guaranteed (see [4] for instance).

Remark 5 *A compromise must be done between reactivity, optimality and*

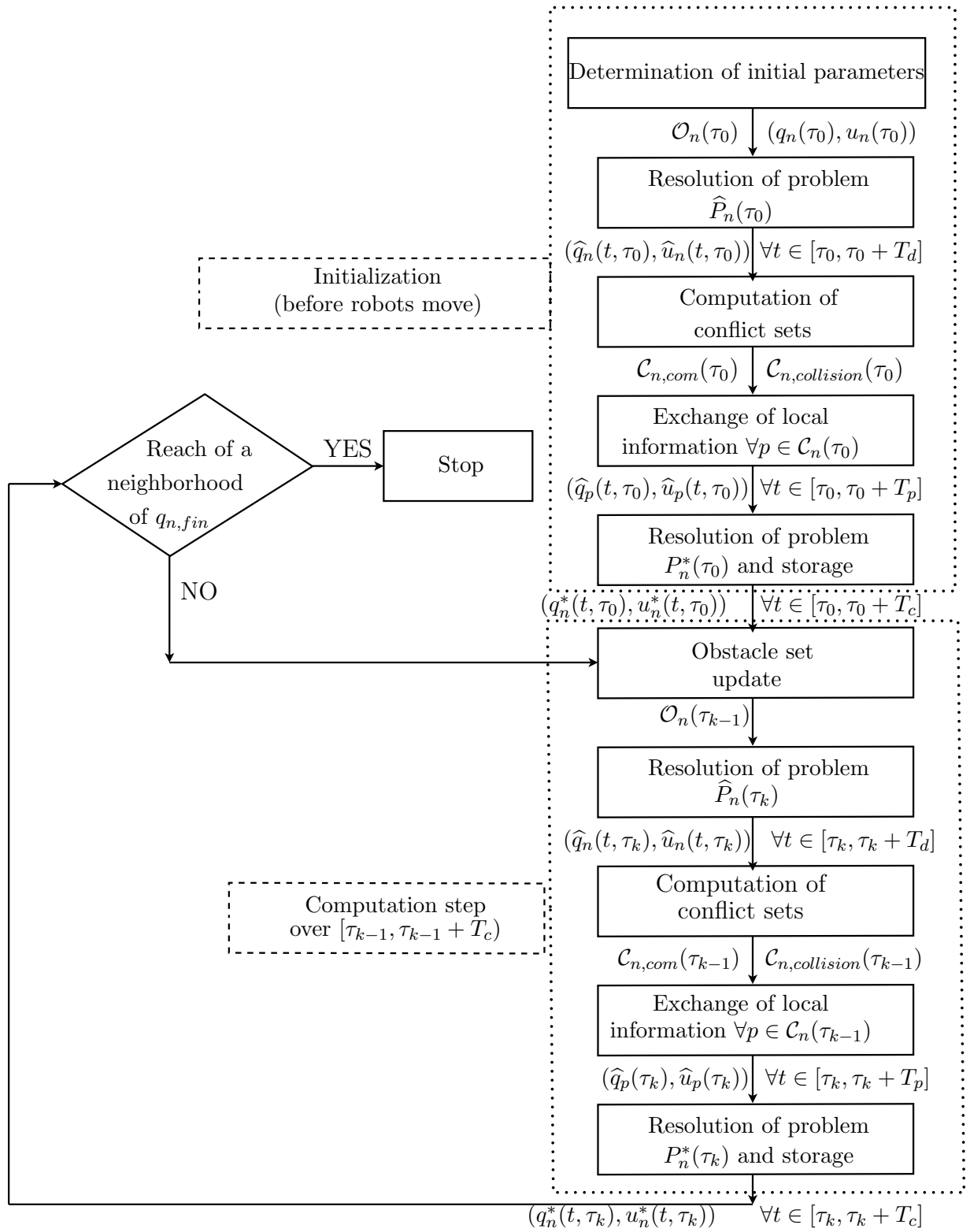


Fig. 5. Flow chart of distributed motion planner

computation time. Indeed, the planning and detection horizon must be sufficiently small in order to have good enough results in terms of computation

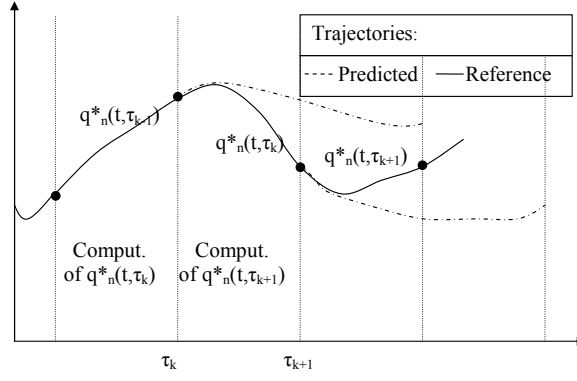


Fig. 6. Implementation of the receding horizon planner

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,
- real time constraint is not ignored,
- quasi-optimal trajectory,
- low communication bandwidth (local information)²

3.3 Technique for solving receding horizon planning problems

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

The key approach is to determine outputs such that equation (1) is mapped to a lower dimensional output space. It will imply that the problem becomes computationally more efficient to solve. Using the flatness property of system

² Since each trajectory and control inputs are approximated using B-spline functions, only control knots and optimal control points are exchanged pairwise (see next part for further details).

(1) (see [9] 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, θ_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) trajectory is planned in this space (see Fig. 7).

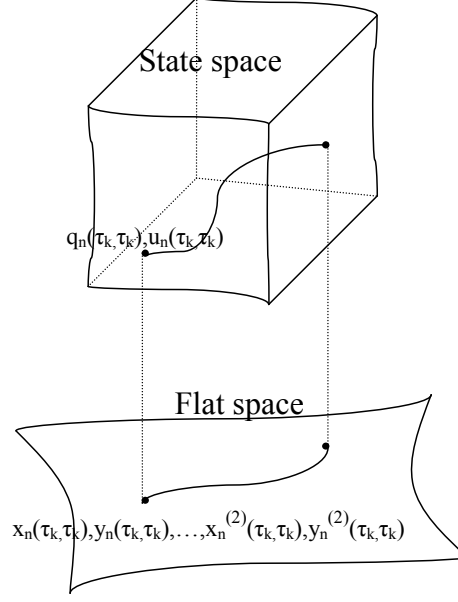


Fig. 7. Flatness and motion planning

Then, in order to transform the optimal trajectory generation problem into a parameter optimization one, a piecewise polynomial function, B-spline, is adopted to specify the trajectory. The B-spline functions are chosen as basis functions due to their flexibility and easiness of enforcing continuity across breakpoints. B-Spline is the 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 $\hat{P}_n(\tau_k)$ (resp. $P_n^*(\tau_k)$), the time interval $[\tau_k, \tau_k + T_d]$ (resp. $[\tau_k, \tau_k + T_p]$) is divided into n_{knot} equal segments with $n_{knot} + 4$ knots to be control knots:

$$nod_0 = \dots = nod_3 = \tau_k < nod_4 < \dots < nod_{n_{knot}+3} = \tau_k + T_d \text{ (resp. } T_p) \quad (23)$$

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

$$\begin{bmatrix} x(t, \tau_k) \\ y(t, \tau_k) \end{bmatrix} = \sum_{j=1}^{3+n_{knot}} C_j B_{j,3}(t) \quad (24)$$

where $C_j \in \mathbb{R}^2$ are the control points and $B_{j,3}$ is the B-spline basis function computed recursively as follows:

$$\begin{aligned}
B_{j,0}(t) &= \begin{cases} 1 & \text{if } \text{nod}_j \leq t < \text{nod}_{j+1} \\ 0 & \text{otherwise} \end{cases} \\
\forall d \in \{1, 2, 3\}, & \\
B_{j,d}(t) &= \frac{t - \text{nod}_j}{\text{nod}_{j+d+2} - \text{nod}_j} B_{j,d-1}(t) + \frac{\text{nod}_{j+d+1} - t}{\text{nod}_{j+d+1} - \text{nod}_{j+1}} B_{j+1,d-1}(t).
\end{aligned} \tag{25}$$

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 [15]. See [17] for a detailed analysis of the efficiency of this approach. To finish, the open-loop control inputs are deduced using equation (22).

4 Simulation results and performance evaluation

In order to highlight the performance of the proposed decentralized strategy, a comparative study with other existing motion planners for a formation of autonomous mobile robots evolving in unknown environments is done. The comparison criteria are: (1) the computation time, (2) the information flow, (3) the easiness of implementation and (4) the optimal traveling time.

4.1 Generalities on existing cooperative motion planners

We have implemented and tested the following approaches suitable for our motion planning problem under the same conditions³:

- 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 [12] 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 to decrease the information flow. From $q_p(\tau_k, \tau_{k-1})$

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

($\forall p \in \mathcal{C}_n(\tau_k)$), robot R_n generates its optimal trajectory $q_n(t, \tau_k)$ and the presumed trajectories $\hat{q}_p(t, \tau_k)$.

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, the presumed and optimal 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 $q_{p,final}$, $\forall p \in \mathcal{C}_n(\tau_k)$.

- a leader / follower receding horizon approach [13] where each robot sequentially computes its optimal trajectory. At 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 who 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 trajectory by taking the already planned trajectories into account. And, so on. The main advantage of this approach is its easiness 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.

5 Simulation results and discussion

The geometrical shape of each robot R_n is included in a circle of radius $\rho_n = 0.2m$. The linear and angular velocities are bounded, i.e. $|v_n| \leq 0.5m.s^{-1}$ and $|w_n| \leq 5rad.s^{-1}$. In the simulation examples, (1) represents the kinematics of R_n . A pentium IV (192Mo of RAM) micro-processor running at 2.4 Ghz operating under linux real time hosts the motion planner written in C.

5.1 Scenario 1

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 fast as possible.

Since the environment is partially known and further explored in real time (i.e. the range of sensors of each robot is of radius $1.5m$), a receding horizon strategy, whose parameters are given in Table 1, is used.

T_p	2s
T_c	0.5s
T_d	2s
ξ	0.25
n_{knot}	3

Table 1

Parameters of the receding horizon planners

Here, we do not consider coupling constraint (5) on maximum separation distance (i.e. $d_{1,com} = d_{2,com} = +\infty$). Thus, only the collision avoidance (6) 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 collision-free trajectory. One can note that, due to the symmetry of this problem, some properties of the studied algorithms can be highlighted⁴.

5.1.1 Centralized and “weakly” decentralized approach [12]

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 $2N(3 + n_{\text{knot}})$. The time needed to find the optimal solution when robots cross over is $172ms$. The simulation results are given in Fig. 8. The linear velocities, shown in Fig. 8(b), are less than $0.5m/s$. 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. Figure 8(c) shows the evolution of the distance between robots. Since it is higher than $0.4m$, the collision avoidance is guaranteed. The time to reach the desired objective is $15.8s$ for robot R_1 and $16.2s$ for robot R_2 .

For the “weakly” decentralized approach [12], a supervisor is not needed. Each robot plans its own trajectory by solving an optimization problem of dimension scale $2(3 + n_{\text{knot}})$ 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 $2N(3 + n_{\text{knot}})$). Therefore, the results obtained are the same as the results using a centralized strategy for this scenario.

⁴ For comparison purpose, 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.

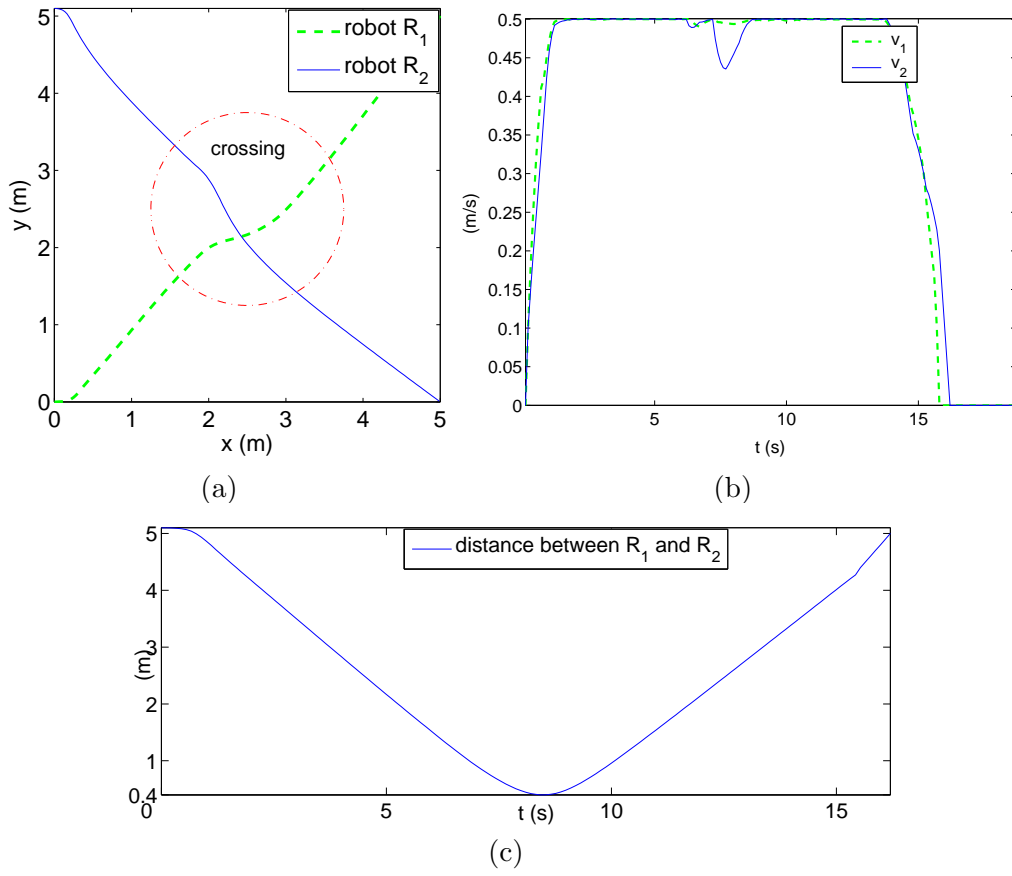


Fig. 8. Scenario 1: centralized and weakly decentralized approaches.

5.1.2 Leader / Follower approach [13]

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 $2(3 + n_{\text{knot}})$ 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 40ms). The simulation results are given in Fig. 9. The linear velocities, shown in Fig. 9(b), are less than 0.5m/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 15.7s for robot R_1 and 16.4s for robot R_2 . Figure 9(c) shows the evolution of the distance between robots. Since it is higher than 0.4m, the collision avoidance is guaranteed.

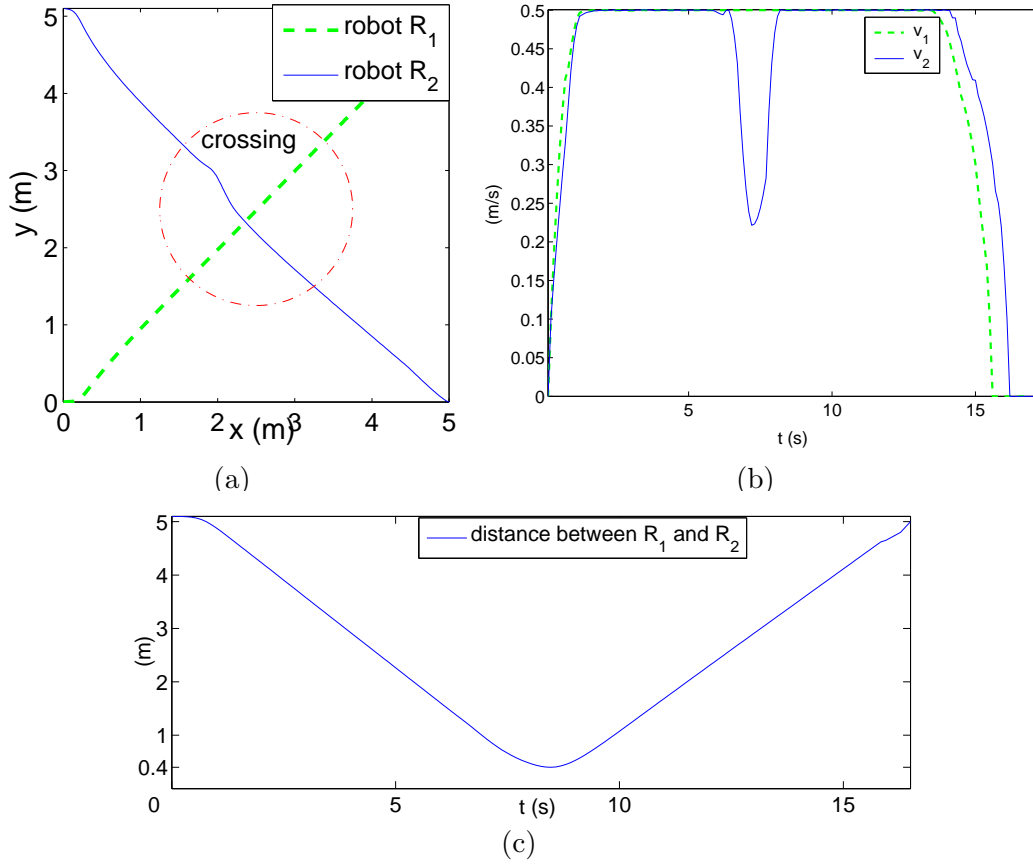


Fig. 9. Scenario 1: leader / follower approach.

5.1.3 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 collision-free trajectory. The time needed to solve the problem when robots cross over is 94ms . The simulation results are given in Fig. 10. The linear velocities, shown in Fig. 10(b), are less than 0.5m/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 16.0s for robot R_1 and 16.3s for robot R_2 . Figure 10(c) depicts the evolution of the distance between robots. Since it is higher than 0.4m , the collision avoidance is guaranteed.

Remark 6 Contrary to the “weakly” decentralized approach [12], constraint (21) is added in order to enforce the degree of correspondence between the optimal and the presumed trajectory known by the neighbors. Thus, the collision avoidance between robots is guaranteed.

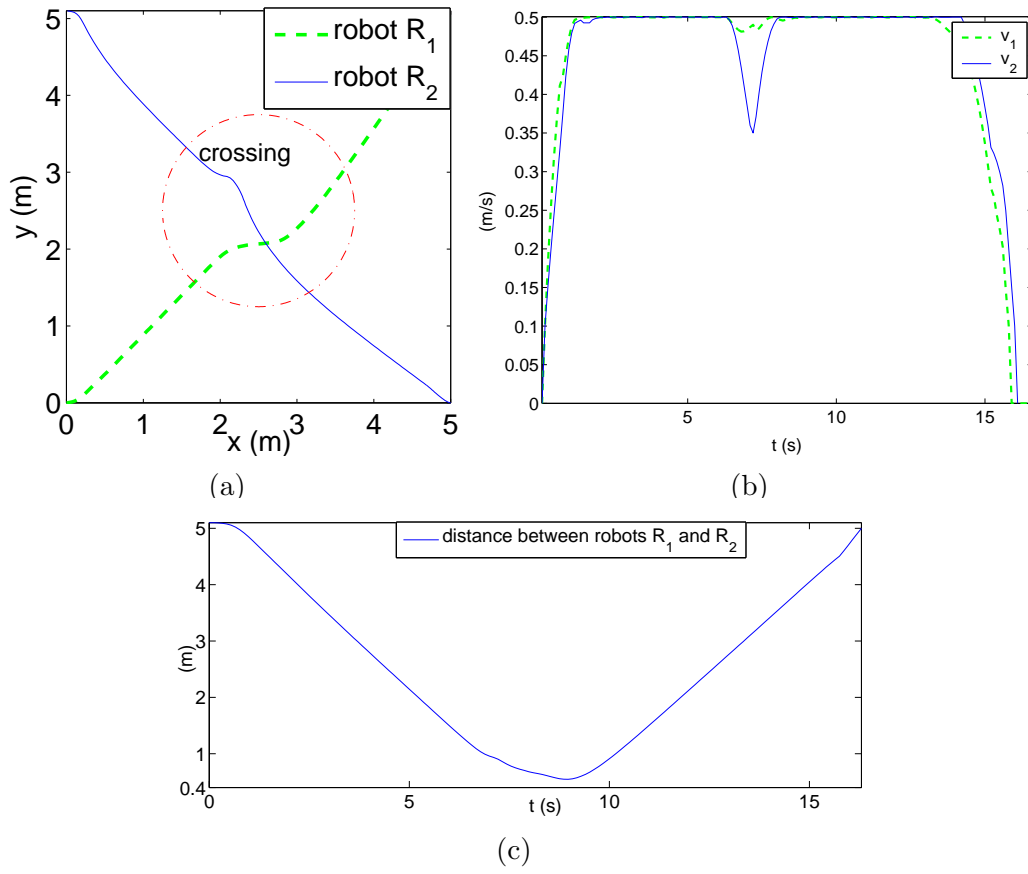


Fig. 10. Scenario 1: proposed decentralized approach.

The comparison in terms of computation time and optimal traveling time for the group using the above algorithms is summarized in Tab. 2.

Approach	Centralized	Leader Follower [13]	“Weakly” decentralized [12]	Proposed decentralized
Maximum computation time	172ms	40ms	172ms	94ms
Optimal traveling time	16.2ms	16.5ms	16.2ms	16.3ms

Table 2

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

In this scenario, the group consists of five robots ($N = 5$) starting at initial configurations given in Tab. 3, with velocities equal to zero. These robots must reach the desired configuration given in Tab. 4 as fast 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_3) , (R_2, R_4) and (R_3, R_5) (see the communication graph given in Fig. 11). The five robots make decisions in order to avoid collision with other robots and obstacles, initially unknown, and to maintain the communication links. The broadcast range of each R_n is $d_{n,com} = 2.5m$.

Robot	x -position (m)	x -position (m)	θ -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 3

Initial configurations (“linear” shape).

Robot	x -position (m)	x -position (m)	θ -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

Table 4

Desired configurations (“triangular” shape).

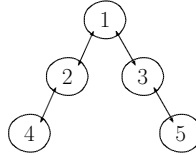


Fig. 11. Communication graph.

Since the environment is partially known (i.e. the range of sensors of each robot is of radius $1.5m$), a receding horizon strategy, whose parameters are given in Table 5, is used. One can note that, due to the number of potential

T_p	2s
T_c	0.5s
T_d	2.5s
ξ	0.25
n_{knot}	3

Table 5

Parameters of the receding horizon planners

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 avoidance⁵, 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 36.5s for the group. Figure 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 0.4m and lower than 2.5m. The other relative distances are given in Fig. 12(c), showing the collision avoidance between robots.

Finally, in order to show the advantages of the proposed decentralized algorithm, the same five vehicle scenario is tested using other algorithms. The comparison in terms of maximum computation time for the design of the optimal trajectory during one planning horizon, optimal traveling time, communication bandwidth and feasibility (i.e. guarantee of collision avoidance and maintenance of communication links) using the algorithms studied is summarized in Tab. 6.

The centralized algorithm provides the best results in terms of optimal 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 online. 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. Therefore, we can conclude that the proposed decentralized algorithm provides the best

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

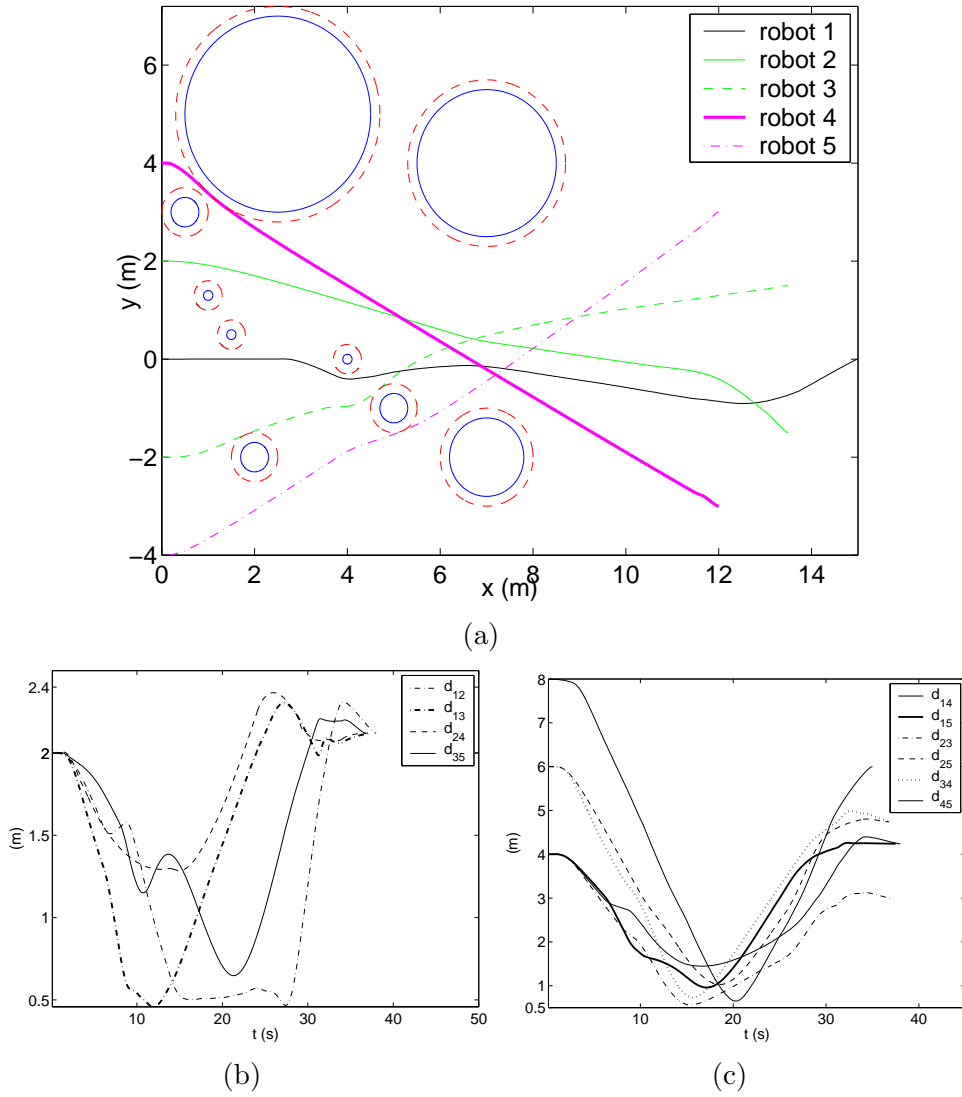


Fig. 12. Scenario 2: proposed decentralized approach.

compromise between optimality and computation time for this scenario.

6 Conclusion

A decentralized algorithm, based on a receding horizon strategy, is proposed in order to solve the motion planning problem for a formation of autonomous mobile robots which is subject to constraints : control bounds, collision avoidance and bounded distances between robots. The main idea is to divide in two steps the generation of the optimal trajectory which satisfies the set of all constraints for each robot. In the first step, an intuitive trajectory is computed which respects just the constraints. In the second step, each robot elaborates its optimal trajectory by taking the notions of intention and engagement of

Approach	Centralized	Leader Follower [13]	“Weakly” decentralized [12]	Proposed decentralized
Maximum computation time	2050ms	313ms	703ms	121ms
Information Flow	global	local	local	local
Feasibility	YES	YES	NO	YES
Optimal traveling time	35s	39s	36s	36.5s

Table 6

Comparison of the proposed decentralized algorithm with other centralized and decentralized algorithms for the scenario 2

robots into account. Simulations on a formation of five mobile robots highlighted the advantages of the proposed strategy in terms of computation time, optimal traveling time, communication bandwidth and feasibility, compared to the best decentralized algorithms of the literature [13], [12]. The most important advantages of the proposed algorithms are the real time implementation, especially when the formation contains more than two robots, and the reduction of the information exchanged between the robots. In fact, each robots requires only local knowledge of its neighbors.

References

- [1] C. Clark, “A coordination platform for multi-robot systems”, *PhD Thesis, Stanford University*, 2004.
- [2] C. Canudas de Wit and O. Sordalen, “Exponential stabilization of mobile robots with nonholonomic constraints”, *IEEE Trans. on Automatic Control*, **37**, No 11, pp. 1791-1797, 1992.
- [3] F. A. Cuzzolaa, J. C. Geromel and M. Mrari, “An improved approach for constrained robust model predictive control”, *Automatica*, **38**, No 7, pp. 1183-1189, 2002.
- [4] M. Defoort, T. Floquet, A. Kokosy and W. Perruquetti, “Integral sliding mode control for trajectory tracking of a unicycle type mobile robot”, *Integrated Computer Aided Engineering*, **13**, pp. 277-288, 2006.

- [5] M. C De Gennaro and A. Jadbabaie, "Formation control for a cooperative multi-agent system using decentralized navigation functions", *American Control Conference*, pp. 14-16, 2006.
- [6] D. V. Dimarogonas and K. J. Kyriakopoulos, "Formation Control and Collision Avoidance for Multi-Agent Systems and a Connection between Formation Infeasibility and Flocking Behavior", *IEEE Conf on Decision and Control*, pp. 84-89, 2005.
- [7] W. Dunbar and R. M. Murray, "Model predictive control of coordinated multi-vehicle formation", *IEEE Conf. on Decision and Control*, 2002.
- [8] W. Dunbar and R. M. Murray, "Distributed receding for multi-vehicle formation stabilization", *Automatica*, pp. 549-558, 2006.
- [9] M. Fliess, J. Lévine, Ph. Martin and P. Rouchon, "Flatness and defect of nonlinear systems: introductory theory and examples", *Int. J. of Control*, **61**, No 6, pp 1327-1361, 1995.
- [10] G. Inalhan, D. Stipanovic and C. Tomlin, "Decentralized optimization, with application to multiple aircraft coordination", *IEEE Conf. on Decision and Control*, 2001.
- [11] T. Keviczky, K. Fregene, F. Borelli, G. Balas and D. Godbole, "Coordinated autonomous vehicle formations: decentralization, control synthesis and optimization", *IEEE American Control Conference*, pp. 2022-2027, 2006.
- [12] T. Keviczky, F. Borelli and G. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems", *Automatica*, **42**, pp. 2105-2115, 2006.
- [13] Y. Kuwata, A. Richards, T. Schouwenaarars and J. P. How, "Decentralized robust receding horizon control", *IEEE American Control Conference*, pp. 2047-2052, 2006.
- [14] J. Lawton, R. Beard and B. Young, "A decentralized approach to formation maneuvers", *IEEE Trans. on Robotics and Automation*, **19**, No 6, pp. 933-941, 2003.
- [15] C. Lawrance, J. Zhou and A. Tits, "User's guide for CFSQP Version 2.5", *Institute for Systems Research, University of Maryland, College Park*.
- [16] D. Mayne, J. Rawlings, C. Rao and P. Scokaert, "Constrained model predictive control: Stability and Optimality", *Automatica*, **36**, No 6, pp. 789-814, 2000.
- [17] M. B. Milam, "Real time optimal trajectory generation for constrained dynamical systems", *PhD Thesis, California Institute of Technology*, 2003.
- [18] N. Miyata and J. Ota and T. Arai and 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**, pp. 769-780, 2002.

- [19] T. Sugar, J. Desai, J. Ostrowski and V. Kumar, “Coordination of multiple mobile manipulators”, *IEEE Conf. on Robotics and Automation*, pp. 3022-3027, 2001.