



HAL
open science

Priority-based intersection management with kinodynamic constraints

Jean Gregoire, Silvère Bonnabel, Arnaud de La Fortelle

► **To cite this version:**

Jean Gregoire, Silvère Bonnabel, Arnaud de La Fortelle. Priority-based intersection management with kinodynamic constraints. 2013. hal-00875435v1

HAL Id: hal-00875435

<https://hal.science/hal-00875435v1>

Preprint submitted on 22 Oct 2013 (v1), last revised 5 May 2014 (v2)

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.

Priority-based intersection management with kinodynamic constraints

Jean Grégoire*

Silvère Bonnabel*

Arnaud de La Fortelle*†

Abstract—We consider the problem of coordinating a collection of robots at an intersection area taking into account dynamical constraints due to actuators' limitations. We adopt the coordination space approach, which is standard in multiple robot motion planning. Assuming the priorities between robots are assigned in advance, and assuming the existence of a collision-free and deadlock-free trajectory satisfying those priorities, we propose a provably safe trajectory planner satisfying kinodynamic constraints. The algorithm is shown to run in real time, to return safe (collision-free) trajectories, and to be efficient enough, in the sense that the optimality gap remains reasonable. Simulation results on synthetic data illustrate the benefits of the approach. Although the paper is devoted to centralized planning, this work opens some avenues for robust distributed multiple robot motion control.

I. INTRODUCTION

A. Motivation

Human error is the sole cause in 57% of all road accidents and is a contributing factor in over 90% [1], [2]. Moreover, traffic congestions motivate the research to improve intersection traffic flow. Intelligent transportation systems are expected to tackle both safety and efficiency issues in the near future. Many systems have been proposed and they have proved their ability to increase traffic efficiency – particularly compared to traffic lights systems – and to reduce the risk of road accidents [3], [4], [5], [6], [7], [8]. Furthermore, more generally, automated conflict management opens new perspectives to improve railway [9] and air transportation systems [10] efficiency.

In transportation systems, safety is usually centralized (e.g. air traffic control, rail management systems) or at least managed locally in a centralized way (e.g. traffic lights). In the future, we anticipate there will be locally full information, e.g. through car-to-car communication being currently standardized. Obviously there will be non-communicating entities, sometimes delays or sensing errors, but our aim is to go from a centralized system in full information down to more reactive schemes, ensuring safety first. Therefore we do not consider here the reactive approach used in lots of distributed systems, not even a deliberative approach based on consensus theory: we want first to plan safe trajectories that are gridlock-free (every vehicles eventually exits the intersection area) and then release some constraints while keeping the safety and efficiency as much as possible.

B. Related work

The standard approach to multi robot motion planning is to decompose the problem into two parts, as initiated in [11]. As presented in [12], the first one consists of determining fixed paths along which robots cross the intersection. The second one consists of computing the velocity profile of each robot along its path: this is a well-known problem studied for applications in automated guided vehicles (AGVs) and robots manipulators.

As first introduced in [13], [14], the path-velocity decomposition enables to introduce an abstract space: the coordination space. It is a standard approach to robot motion planning [15], [16], and the motion planning problem in the real space boils down to finding an optimal trajectory in the coordination space that is collision-free with respect to an obstacle region. The coordination space is a n -dimensional space (where n denotes the number of robots in the intersection) and the obstacle-region has a cylindrical structure [17]. In [12], [18], we have revisited the notion of priorities to propose a novel framework for automated intersection management based on priority assignment. It is a very intuitive notion: the priority graph indicates the relative order of robots. Our framework enables to decompose the motion planning problem in the coordination space into a combinatorial problem: priority assignment and a continuous problem: finding an optimal/efficient trajectory with assigned priorities.

The ambition of this framework is to enable more robustness and distribution in future automated intersection management systems. Indeed, existing intersection management systems such as proposed in [6], [3], [8] plan the complete trajectories of robots through the intersection and ensuring safety requires robots to follow precisely the planned trajectory. By contrast, if priorities only are planned, the priority graph can be conserved even if some unpredictable event requires a robot to slow down for some time. Moreover, trajectory planning at every time-step can be distributed over robots if every robot has knowledge of the assigned priorities.

It is now clear that the combinatorial problem of assigning judicious priorities is inherently difficult, as noticed in [19] and developed in the priority-based framework in [18]. As a result, we will only consider in the present paper the issue of planning "good" trajectories for fixed in advance priorities. When the robots can start and stop instantaneously, it is relatively easy to define an optimal trajectory for fixed priorities. This trajectory is referred to as the left-greedy trajectory [19], [18]. However, replacing this with acceler-

* Mines ParisTech, Centre de Robotique, Mathématiques et Systèmes, 60 Bd St Michel 75272 Paris Cedex 06, France

† Inria Paris - Rocquencourt, IMARA team, Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay, France

ation (and higher derivatives) bounds turns the optimization problem into a "highly non-trivial" problem (as suggested in the conclusion of the paper [19]). In the present paper, we address the challenging problem of defining safe and efficient (although suboptimal) trajectories that respect this type of constraints. In [20], the problem is formulated as a mixed integer nonlinear programming problem, and the solution proposed is suitable only for a "reasonable" and fixed number of robots. Moreover, priority assignment and trajectory planning are not decoupled. In the present paper, we focus on a low complexity solution to the trajectory planning problem with assigned priorities which is applicable for a large and potentially varying number of robots.

C. Contributions

We introduce a theoretical tool: the braking trajectory, which is a virtual trajectory obtained letting all robots slowing down as much as possible to stop. The key idea of the paper is to ensure that at every time-step, the (virtual) braking trajectory is collision-free. To this respect, it can be related to the approach of [7]. However, in the present paper, the notion of priorities is a geometric concept taking advantage of the topology of the coordination space (see [18]). It is more general than and quite different from the relation between robots of [7] for the following reasons. First of all, the relation of [7] is an order whereas the priority relation is not necessarily an order [18]. Secondly, in [7], the trajectory of the robots and the priority assignment policy are both maximally aggressive. At high traffic density, it is not necessarily efficient because priorities should be assigned considering queues upstream the intersection. By contrast, in this paper, the priority assignment policy can be of any kind. Moreover, the dynamic constraints considered presently are more general than bounded acceleration and we just assume in our work that the set of reachable positions from any state of the system is bounded. Finally, the decomposition of the problem using the priority graph highlights that the efficiency of the priority assignment policy should be distinguished from the efficiency of the trajectory planning with assigned priorities.

With the proposed planner, robots are maximally aggressive, i.e. always maximize the distance travelled at every time-step. However, they do not accelerate if the virtual braking trajectory becomes unsafe or violates a priority, i.e. they ensure the existence of a failsafe maneuver for the system of robots at any time. We present a trajectory planner with assigned priorities that consists of just-in-time braking. The proposed algorithm does not return an optimal trajectory, in terms of averaged travel time over robots. However, we provide a theoretical result highlighting that the optimality gap is due to (positive) acceleration constraints and is of the order of the time necessary to go from stop to full speed. Finally, simulation results confirm the safety and the efficiency of the approach.

Section II and III present the modelling assumptions and recall the basics of the priority-based framework of [12], [18]. Section IV introduces the motion planner algorithm

along with its safety and efficiency guarantees. Finally, simulation results of Section V illustrate the efficiency of the approach.

II. MODELLING ASSUMPTIONS

A. Fixed paths assumption and the coordination space

We assume that robots are constrained to follow predefined paths to go through the intersection. The paths are not necessarily straight lines: robots are just considered as driving along fixed tracks. This can be achieved by a low-level controller. This standard assumption [13], [21], [22], [23], [7] fits well intersections in a road network, where robots travel along lanes.

Assumption 1 (Path following). *Every robot i follows a particular path γ_i and we denote $x_i \in \mathbb{R}$ its curvilinear coordinate along the path. The configuration of the system of robots is $x = (x_i)_{i \in \{1 \dots n\}}$ and we denote $x(t)$ the evolution of x through time $t \in [0, T]$.*

Figure 1 illustrates the path following assumption.

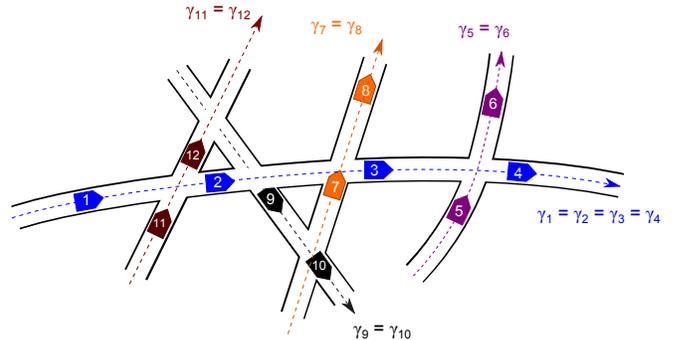


Fig. 1. The path following assumption. All robots in the same lane (depicted with the same color) travel along the same geometric path with independent velocity profiles.

The curvilinear coordinates are normalized, so that $x \in \chi = [0, 1]^n$ where n denotes the number of robots going through the intersection (possibly changing through time). The configuration space χ is known as the coordination space [24], [17], [13]. In the rest of the paper, $\{e_i\}_{1 \leq i \leq n}$ denotes the canonical basis of χ .

B. The obstacle region

The use of the coordination space and the results of this paragraph are standard [17]. As every robot occupies a non-empty geometric region, some states must be excluded to avoid collisions between robots.

Definition 1 (Obstacle region, Obstacle-free region). *The obstacle region χ_{obs} is the open set of all collision configurations. $\chi_{\text{free}} = \chi \setminus \chi_{\text{obs}}$ denotes the obstacle-free space.*

A collision occurs when two robots occupy a common region of space, so that:

Property 1 (Cylindrical structure). *The obstacle region can be described as the union of $n(n-1)/2$ open cylinders χ_{obs}^{ij} corresponding to as many collision pairs: $\chi_{\text{obs}} = \cup_{i>j} \chi_{\text{obs}}^{ij}$.*

Figure 2 displays the obstacle region and a collision configuration for a two-path intersection.

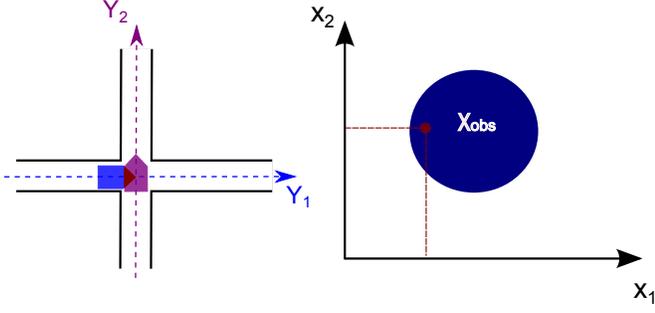


Fig. 2. The left drawing depicts two robots in collision in a 2-path-intersection. The right drawing depicts the corresponding configuration in the coordination space that belongs to the obstacle region.

Assumption 2 (Cylinders convexity). *Each cylinder χ_{obs}^{ij} has an open bounded convex cross-section (in the plane generated by e_i and e_j).*

C. Kinodynamic constraints

To perform motion planning optimization, we need to make hypothesis on the technical constraints of the robots. These include kinematic constraints (maximum velocity, maximum curve radius, etc.) and dynamic constraints (limited acceleration, adherence, jerk, etc.).

Definition 2 (Robots constraints). *Let p denote the degree of the constraints and n the number of robots. Let $s(t) = (x \ x' \ \dots \ x^{(p)})(t) \in \mathbb{R}^{n \times (p+1)}$ denote the state of the system. We let $x(t) = \pi(s(t))$ denote the first column of the state $s(t)$, that is the position of all robots.*

We say a trajectory x respects the kinodynamic constraints C if: $\forall i \in \{1 \dots n\}, \forall t \in [0, T]$ such that $x_i(t) \in (0, 1)$, we have $s_i(t) \in C_i$ with $C_i \subset \mathbb{R}^{p+1}$ representing the constraints for robot i and $C = \prod_{i \in \{1 \dots n\}} C_i \subset \mathbb{R}^{n \times (p+1)}$.

Note that because of the boundedness condition on χ , the constraints must be respected for any time t such that $x(t)$ is not at the boundary of χ (i.e. $x_i(t) \in (0, 1)$). This ensures that x can be artificially stopped at the boundary of χ to stay inside χ . Note also that every robot can have different constraints C_i , and C_i can not necessarily be expressed in a product form (for example, the constraint on the acceleration can depend on the velocity).

Assumption 3 (Boundedness and continuity of reachable positions). *The set of reachable positions from state s_0 in a time-length t :*

$$\chi_{\text{reach}}(s_0, t) = \left\{ x(t) \left| \begin{array}{l} x \text{ respects the constraints } C \\ s(0) = s_0 \end{array} \right. \right\} \quad (1)$$

is continuous with respect to s_0 and is a bounded hypercube of $x_0 + \mathbb{R}_+^n$.

Note that, the above assumptions imply in particular that:

- 1) robots cannot travel backwards in the intersection,

- 2) and from a given state s_0 , the set of reachable positions in finite time is bounded, and the bounds depend on the state s_0 of the robots (position, velocity, acceleration, etc.).

III. THE PRIORITY-BASED FRAMEWORK

In this section, we recall the basics of priority-based intersection management introduced in our previous work [12], [18].

A. The priority relation

Consider the region $\chi_{\text{obs}}^{i>j}$ defined as follows and depicted in Figure 3:

$$\chi_{\text{obs}}^{i>j} = \chi_{\text{obs}}^{ij} - \mathbb{R}_+ e_i + \mathbb{R}_+ e_j \quad (2)$$

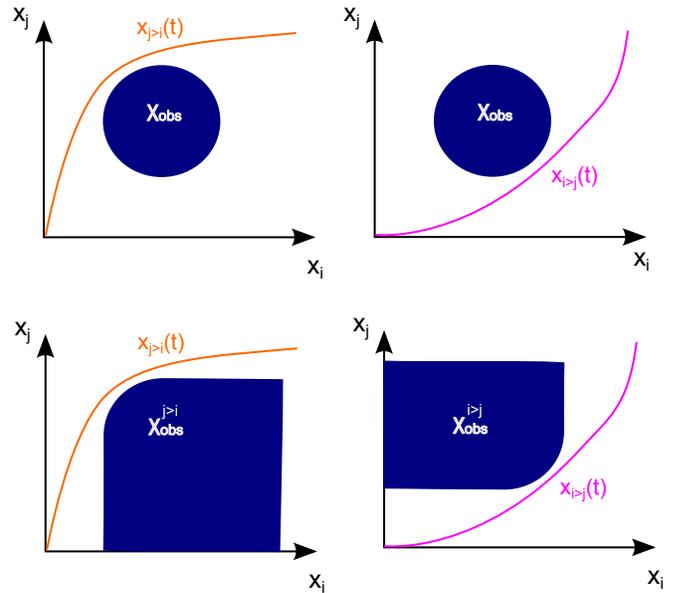


Fig. 3. The top drawings represent in the plane (x_i, x_j) the obstacle region χ_{obs} , a feasible trajectory $x_{i>j}$ respecting priority $i > j$ and a feasible trajectory $x_{j>i}$ respecting priority $j > i$. The bottom drawings depict $\chi_{\text{obs}}^{j>i}$ and $\chi_{\text{obs}}^{i>j}$.

We define a natural binary relation corresponding to priority relations between robots.

Definition 3 (Priority relation). *An exit trajectory x induces a binary relation \succ on the set $\{1 \dots n\}$ as follows. For $i \neq j$ s.t. $\chi_{\text{obs}}^{ij} \neq \emptyset$, $i \succ j$ if x is collision-free with $\chi_{\text{obs}}^{i>j}$.*

B. The priority graph

The priority relation can be described by a graph G with nodes $\{1 \dots n\}$, where each edge represents the relative priority of a pair of robots.

Definition 4 (Priority graph). *Given a feasible trajectory x , the priority graph is the oriented graph G whose vertices are $\{1 \dots n\}$ and such that $i \xrightarrow{G} j$ if $i \succ j$.*

An example of a priority graph for 3 robots along 3 distinct paths is described in Figure 4.

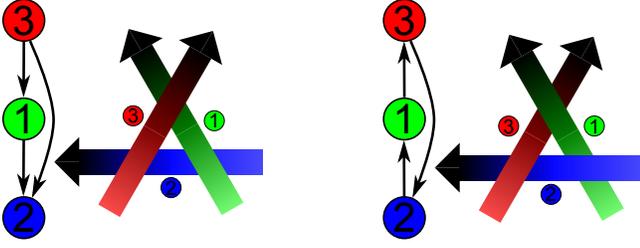


Fig. 4. Two representations of priority relations. Robots along a path in foreground have priority over robots along a path in background.

C. Problem formulation

The initial state of the robots is s^{init} , and the goal region is $\chi_{\text{goal}} = \{\mathbf{1} = (1 \cdots 1)\} \subset \chi_{\text{free}}$.

Definition 5 (Feasible trajectory, Exit trajectory). *A feasible trajectory for the considered problem is a trajectory $x : [0, T] \rightarrow \chi_{\text{free}}$ respecting constraints C and such that $s(0) = s^{\text{init}}$.*

If $x(T) \in \chi_{\text{goal}}$, x is an exit trajectory.

The optimality problem consists of finding an exit trajectory x^* that minimizes the average time spent in the intersection, i.e. that minimizes the cost function $c(x)$ defined below.

Definition 6 (Cost function). *Given an exit trajectory x , the cost function c associates x with the real positive value $c(x)$ defined below:*

$$c(x) = \frac{1}{n} \sum_{i=1}^n T_i = \frac{1}{n} \sum_{i=1}^n x_i^{-1}(1) \quad (3)$$

with $x_i^{-1}(x_i^0)$ denoting the first date at which x_i reaches x_i^0 and $T_i = x_i^{-1}(1)$ being the exit time for robot i .

IV. MOTION PLANNER WITH ASSIGNED PRIORITIES

In this section, we focus on the solution of the problem when priorities are assigned in advance. The key idea is that if robots wait to be at the boundary of the collision region to brake (as it is the case without dynamic constraints in [12]), collisions will occur because robots cannot stop instantly. That is why we need to anticipate the approach of the collision region. This can be done introducing two virtual trajectories as follows.

A. Introducing maximal and minimal trajectories

Definition 7. *The minimal (resp. maximal) trajectory from state s^0 , denoted $\underline{x}_{|s^0}$ (resp. $\bar{x}_{|s^0}$), are defined below:*

$$\begin{aligned} \underline{x}_{|s^0}(t) &= \min_{\chi_{\text{reach}}}(s_0, t) \\ \bar{x}_{|s^0}(t) &= \max_{\chi_{\text{reach}}}(s_0, t) \end{aligned}$$

One can view the minimal trajectory as a braking trajectory, and the maximal trajectory as an accelerating trajectory.

The concepts are illustrated by Figure 5 where the kinodynamic constraints have the special following form:

$$C_i^{\text{acc}} = \left\{ (x_i, x'_i, x''_i) \mid \begin{array}{l} 0 \leq x'_i \leq v_i^{\text{max}} \\ a_i^{\text{min}} \leq x''_i \leq a_i^{\text{max}} \end{array} \right\} \quad (4)$$

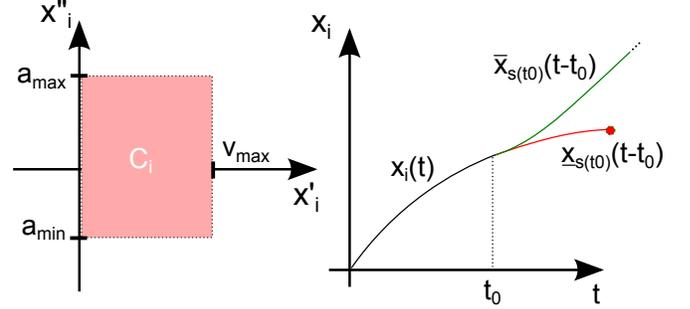


Fig. 5. The left drawing depicts an example of kinodynamic constraints where robots have uniform minimal/maximal velocity and acceleration along their paths. The right drawing depicts the corresponding minimal/maximal trajectories $\underline{x}_{s(t_0)}$ and $\bar{x}_{s(t_0)}$.

Before proposing a motion planner based on those virtual trajectories, a last natural assumption must be made: an early braking (resp. acceleration) implies a delay (resp. advance) that can never be caught up.

Assumption 4 (Benefit of early braking/acceleration). *Say $t^1 > t^0$ and assume that $\forall t \in (t^0, t^1), x(t) > \underline{x}_{|s(t^0)}(t)$. Then, for $t^2 > t^1$, $\underline{x}_{|s(t^1)}(t^2) > \underline{x}_{|s(t^0)}(t^2)$.*

Similarly, assume that $\forall t \in (t^0, t^1), x(t) < \bar{x}_{|s(t^0)}(t)$. Then, for $t^2 > t^1$, $\bar{x}_{|s(t^1)}(t^2) < \bar{x}_{|s(t^0)}(t^2)$.

B. The motion planner

The time t is first discretized, and the trajectory of the robots $x(t)$ is computed iteratively as described in the following Algorithm 1. Indeed, at every time-step t , the trajectory up to time $t + \Delta T$ can be computed as follows:

- Cycling through all robots, we select a particular robot i (line 5)
- we compute a trajectory \tilde{s} defined letting this robot accelerate as much as possible and letting all other robots decelerate as much as possible (lines 6-11)
- we define a complete virtual trajectory χ^{virtual} concatenating $\pi(\tilde{s})$ up to time $t + \Delta T$, and then letting all vehicles brake as much as possible at time $t + \Delta T$ (line 13)
- If this virtual path is such that no collision and no priority violation occurs, it means there exists a failsafe maneuver such that robot i accelerates as much as possible, and we let it do so. Otherwise, robot i must brake (lines 14-18). Thus, at each time-step t , each robot i exclusively follows its maximal $\bar{x}_{i|s(t)}$ or minimal trajectory $\underline{x}_{i|s(t)}$ in the next time-step.

The defined trajectory thus appears as a natural extension of the left-greedy trajectory introduced in [19], in the sense that in the absence of kinodynamic constraints ($p = 1$), it coincides with it. Indeed, in this case the robots can stop

Algorithm 1 The motion planner with assigned priorities

Input: s^{init} , feasible priority graph G

function MAXIMALLYAGGRESSIVETRAJECTORY

$T \leftarrow 0$

$s(0) \leftarrow s^{\text{init}}$

while $x(T) \notin \chi_{\text{goal}}$ **do**

5: **for** $i \in \{1..n\}$ **do**

for $t \in [0, \Delta T]$ **do**

for $j \neq i$ **do**

$\tilde{s}_j(t) \leftarrow \underline{s}_{j|s(T)}(t)$

end for

10: $\tilde{s}_i(t) \leftarrow \bar{s}_{i|s(T)}(t)$

end for

$s^{\text{f}} = \tilde{s}(\Delta T)$

$\chi^{\text{virtual}} \leftarrow \pi(\tilde{s}([0, \Delta T])) \cup \underline{x}_{|s^{\text{f}}}(\mathbb{R}_+)$

if $\exists j \xrightarrow{G} i$ s.t. $\chi^{\text{virtual}} \cap \chi_{\text{obs}}^{j>i} \neq \emptyset$ **then**

15: $s_i(T + \Delta T) \leftarrow \underline{s}_{i|s(T)}(\Delta T)$

else

$s_i(T + \Delta T) \leftarrow \bar{s}_{i|s(T)}(\Delta T)$

end if

end for

20: $T \leftarrow T + \Delta T$

end while

return $(x(t))_{t=0..T}$

end function

instantly and the block from Line 6 to Line 18 simply consists of checking that maximum speed during the next time-step is safe: if it is not the case the robot is stopped. Note also that if the state $s(t)$ is such that the braking trajectory $\underline{x}_{|s(t)}$ is collision-free, the state $s(t)$ is not an "Inevitable Collision State" (ICS), as defined in [25]. Indeed, $\underline{x}_{|s(t)}$ is a trajectory that enables to avoid any collision, i.e. $s(t)$ is not an ICS. Finally, note that the block from Line 6 to Line 18 can be distributed over robots if they all have knowledge of the priority graph and other robots' state. However, distributed control is not the focus of the present paper.

C. Safety guarantees

The theorem below exhibits the safety guarantee provided by the proposed motion planner.

Theorem 1 (Safety guarantees). *Assume that there exists some exit trajectory respecting priorities defined by G and the initial state s^{init} is such that the initial braking trajectory $\underline{x}_{|s^{\text{init}}}$ is collision-free, Algorithm 1 terminates and returns a collision-free exit trajectory, for sufficiently small ΔT .*

Proof. It is assumed that there exists some exit trajectory respecting priorities defined by G , so that G is feasible as defined in [18], and the trajectory cannot reach a deadlock configuration (for sufficiently small ΔT). Until χ_{goal} is reached, at any time there is at least one robot at a coordinate $x_{i_0} < 1$ moving forward. Indeed if this was not true it would mean that the robots have reached a deadlock configuration.

There is thus a lower bound, say μ , for the distance travelled by some of the robots in a time-length ΔT , depending on the constraints. This implies x necessarily reaches χ_{goal} in finite time (of order at most $O(n/\mu)$) and Algorithm 1 terminates.

Moreover, for any priority $j \xrightarrow{G} i$, the braking trajectory from any state of the returned trajectory is collision-free with respect to $\chi_{\text{obs}}^{j>i}$ by construction. At every time-step, there are two options for the robot with lower priority:

- either i brakes as much as possible. The braking trajectory is the worst-case scenario for collision with respect to $\chi_{\text{obs}}^{j>i}$ because $\chi_{\text{obs}}^{j>i} + \mathbb{R}_+ \mathbf{e}_i \subset \chi_{\text{obs}}^{j>i}$ (see Fig. 3) and $\chi_{\text{obs}}^{j>i}$ is a cylinder. Since the braking trajectory is collision-free with respect to $\chi_{\text{obs}}^{j>i}$, the returned trajectory is necessarily also collision-free with respect to $\chi_{\text{obs}}^{j>i}$.
- or i accelerates as much as possible. For the same reasons as previously, \tilde{x} is the worst-case scenario for collision with respect to $\chi_{\text{obs}}^{j>i}$. Since \tilde{x} is collision-free with respect to $\chi_{\text{obs}}^{j>i}$, the returned trajectory is also collision-free with respect to $\chi_{\text{obs}}^{j>i}$.

As a result, x is collision-free at every time-step and reaches χ_{goal} : it is a collision-free exit trajectory. \square

D. Discussion on efficiency

The motion planner presented in this paper does not return an optimal trajectory with regards to the objective (3), whose purpose is to minimize the average time spent in the intersection, and finding an optimal trajectory remains a challenging problem whenever dynamical constraints come into play [19]. However, the following theorem allows to approximately evaluate the optimality gap. Indeed, it shows that suboptimality of the algorithm has two sources: first some time is lost picking a trajectory such that at any time the minimal trajectory is collision-free. This is not an issue, as it endows the algorithm with robustness properties, since it means that an emergency stop can occur at any time without generating any collision. Then, some time is also lost during the (positive) acceleration phases.

Theorem 2 (Efficiency guarantee). *Assume that there exists some exit trajectory respecting priorities defined by G and the initial state s^{init} is such that the initial braking trajectory $\underline{x}_{|s^{\text{init}}}$ is collision-free. Relaxing only the positive acceleration constraints, assume robots can reach maximum speed instantaneously at any time. This implies the maximal trajectories originating from two initial states s^0 and s^1 with the same position coincide:*

$$\pi(s^0)_i = \pi(s^1)_i \Rightarrow \bar{s}_{i|s^0} = \bar{s}_{i|s^1} \quad 1 \leq i \leq n \quad (5)$$

Define robust exit trajectories as exit trajectories x such that for all $t \in [0, T]$ the braking trajectory from state $s(t)$ is collision-free. The collision-free exit trajectory returned by Algorithm 1 is an optimal robust exit trajectory as $\Delta T \rightarrow 0$.

Proof. Let x^* denote the trajectory returned by Algorithm 1 and \tilde{x} a robust exit trajectory. Since robots can reach full

speed instantaneously, the trajectory \tilde{x} can surpass x^* at time $T^0 + \Delta T$ (i.e. for the first time, a component i of \tilde{x} becomes strictly greater than the same component of x^*) if and only if component i of x^* is following the braking trajectory for $t \in [T^0, T^0 + \Delta T]$. If the braking trajectory is followed, it implies that the final configuration of the braking trajectory from state $s^*(T^0 + \Delta T)$ is very close to some forbidden region $\chi_{\text{obs}}^{j>i}$, arbitrarily close to the frontier of the forbidden region as $\Delta T \rightarrow 0$. As a consequence, surpassing \tilde{x} would result in a small increase in the final configuration of the braking trajectory from state $\tilde{s}(T^0 + \Delta T)$, the frontier would be crossed and the braking trajectory would not be collision-free. In conclusion, the collision-free exit trajectory returned by Algorithm 1 is an optimal robust exit trajectory. \square

The fact that Algorithm 1 is not robust-optimal when maximum speed can not be reached instantly is easy to understand. Indeed, according to the algorithm, the robots brake only when absolutely necessary to avoid a collision or a priority violation. This leads some robots to stop and restart, whereas they could have anticipated, and lowered their speed a little so that the re-acceleration phase would be shorter after having let the priority robots pass first. In the presence of positive acceleration constraints as well, we thus have an idea of the optimality gap which appears to be at most of the order of the time necessary for the slowest robot to go from stop to maximum speed. In particular, for $p = 2$ this implies the optimality gap is of order $O(v^{\max}/a^{\max})$.

V. SIMULATIONS

The algorithms presented in this paper have been implemented into a simulator coded in Java. Our algorithms have proved their ability to run in real-time.

A. Setting and results

Only straight paths are implemented (for simplicity's sake) and all robots are supposed to be circle-shaped with a common radius R . The kinodynamic constraints of the robots concern only the maximal velocity and minimal/maximal acceleration. Moreover, all robots are supposed to have identical kinodynamic constraints.

$$\forall i \in \{1 \dots n\}, C_i^{\text{acc}} = \left\{ (x_i, x'_i, x''_i) \mid \begin{array}{l} 0 \leq x'_i \leq v^{\max} \\ a^{\min} \leq x''_i \leq a^{\max} \end{array} \right\}. \quad (6)$$

In the simulation results presented in this section, we take as priority assignment policy the maximally aggressive priority assignment policy that consists for every robot of taking priority over another robot when it approaches the conflicting region. This priority assignment policy can lead to deadlock configurations (see [18]), but with a very small probability in case of low traffic density as in the presented simulations. This policy is used for the sake of simplicity, the priority assignment policy not being the focus of this paper.

Simulations have been carried out for the 4-path-intersection depicted in Figure 7. At full speed, the distance travelled in one time-step is R and at full acceleration, 20 time-steps are required for the robots to reach full speed.

Figure 6 depicts the increase in travel time for different traffic densities. The increase in travel time is the delay due to coordination, i.e. the difference with the ideal travel time which is the travel time of robots in the absence of other robots. It is expressed in percentage of the ideal travel time. The increase in travel time vanishes as the density approaches 0 since it becomes very unlikely that they need to coordinate to avoid collisions. The traffic density in percentage is the ratio between the actual traffic density and the maximum traffic density (continuous flow of robots). The robots are generated randomly at a constant rate over time. The video of the simulation for a traffic density of 10% is available at <http://youtu.be/bJHdf3Ab11I>.

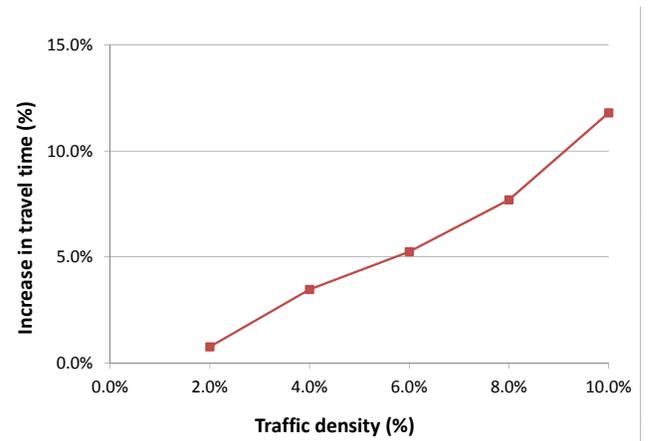


Fig. 6. Simulation results: plot of the averaged increase in travel time against the traffic density for the intersection of Figure 7

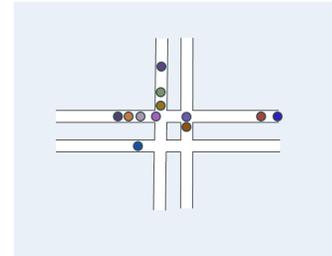


Fig. 7. The 4-path-intersection used for simulations

B. Comments

First of all, our algorithm succeeds to work in real time, and one can observe in simulations (notably on the video) that collisions never occur. This confirms the fact that the planner guarantees safety under dynamic constraints. One can see in Figure 6 that at a traffic density of 10% on each path, the increase in travel time due to coordination to avoid other robots is less than 15% which seems a low price to pay to ensure safe coordination. Note that we do not present simulation results at higher traffic densities because it would require to define a more complex priority assignment policy (at least to avoid deadlocks), which is a challenge in itself, and beyond the scope of the present paper.

VI. CONCLUSIONS AND DISCUSSION

The results presented in this paper prove that when priorities are assigned, it is possible to plan a safe and quite efficient trajectory respecting the priority graph and the dynamic constraints of the robots. The use of the braking trajectory enables to anticipate the need to brake just-in-time, and as a byproduct provides robustness guarantees. The returned trajectory is not optimal but the optimality gap is of the order of the time required for a robot to go from stop to full speed, which seems reasonable.

If robots have knowledge of the priority graph and other robots' state, the motion planner can be distributed over robots. Moreover, if the robots drift from the planned trajectory but if no priority has been violated, it is possible to run the motion planner from a new initial state to get a new exit trajectory respecting the assigned priorities. This opens avenues for robust distributed multiple robot motion control.

REFERENCES

- [1] J. Treat, N. Castellan, R. Stansifer, R. Mayer, R. Hume, D. Shinar, S. McDonald, and N. Tumbas, *Tri-level Study of the Causes of Traffic Accidents: Final Report. Volume I: Causal Factor Tabulations and Assessments*, 1977. [Online]. Available: http://books.google.gp/books?id=_DTHMQAACAAJ
- [2] NCSA, "National center for statistics and analysis, traffic safety facts 2003," U.S. DOT, Washington, DC, Tech. Rep., 2004.
- [3] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, July 2004, pp. 530–537.
- [4] —, "Mitigating catastrophic failure at intersections of autonomous vehicles," in *AAMAS Workshop on Agents in Traffic and Transportation*, Estoril, Portugal, May 2008, pp. 78–85.
- [5] A. Colombo and D. D. Vecchio, "Efficient algorithms for collision avoidance at intersections," *Hybrid Systems: Computation and Control*, 2012.
- [6] I. Zohdy and H. Rakha, "Optimizing driverless vehicles at intersections," in *10th ITS World Congress Vienna, Austria*, October 2012.
- [7] H. Kowshik, D. Caveney, and P. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 804–818, March 2011.
- [8] O. Mehani and A. De La Fortelle, "Trajectory planning in a crossroads for a fleet of driverless vehicles," in *Proceedings of the 11th international conference on Computer aided systems theory*, ser. EUROCAST'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 1159–1166. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1783034.1783191>
- [9] Ismail and Sahin, "Railway traffic control and train scheduling based on inter-train conflict management," *Transportation Research Part B: Methodological*, vol. 33, no. 7, pp. 511–534, 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0191261599000041>
- [10] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: a study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, Apr 1998.
- [11] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.
- [12] J. Gregoire, S. Bonnabel, and A. De La Fortelle, "Optimal cooperative motion planning for vehicles at intersections," in *Navigation, Perception, Accurate Positioning and Mapping for Intelligent Vehicles, Workshop, 2012 IEEE Intelligent Vehicles Symposium*, 2012.
- [13] S. Leroy, J. P. Laumond, and T. Simeon, "Multiple path coordination for mobile robots: A geometric algorithm," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999, pp. 1118–1123.
- [14] S. LaValle and S. Hutchinson, "Optimal motion planning for multiple robots having independent goals," in *Proceedings of the IEEE International Conference on Robotics and Automation, 1996*, vol. 3, Apr 1996, pp. 2847–2852 vol.3.
- [15] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.
- [16] T. Lozano-Perez, "Spatial planning: A configuration space approach," 1980.
- [17] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at <http://planning.cs.uiuc.edu/>.
- [18] J. Gregoire, S. Bonnabel, and A. De La Fortelle, "Priority-based coordination of robots," Jun. 2013, 29 pages. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00828976>
- [19] R. Ghrist and S. M. Lavalle, "Nonpositive curvature and pareto optimal coordination of robots," *SIAM Journal on Control and Optimization*, vol. 45, pp. 1697–1713, November 2006. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1237592.1237600>
- [20] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 295–310, 2005.
- [21] T. Fraichard and C. Laugier, "Planning movements for several coordinated vehicles," in *IEEE/RSJ International Workshop on Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89.*, Sep 1989, pp. 466–472.
- [22] C. Hafner, Cunningham and D. Vechhio, "Automated vehicle-to-vehicle collision avoidance at intersections," 2011.
- [23] S. Akella and S. Hutchinson, "Coordinating the motions of multiple robots with specified trajectories," in *Proceedings of the IEEE International Conference on Robotics and Automation, 2002. ICRA '02.*, vol. 1, 2002, pp. 624–631 vol.1.
- [24] P. O'Donnell and T. Lozano-Periz, "Deadlock-free and collision-free coordination of two robot manipulators," in *Proceedings of the IEEE International Conference on Robotics and Automation, 1989.*, May 1989, pp. 484–489 vol.1.
- [25] T. Fraichard and H. Asama, "Inevitable collision states - a step towards safer robots?" *Advanced Robotics -Utrecht-*, vol. 18, no. 10, pp. 1001–1024, 2004. [Online]. Available: <http://hal.inria.fr/inria-00182063>