Priority-based intersection management with kinodynamic constraints

Jean Gregoire, Silvère Bonnabel, Arnaud De La Fortelle

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 actuator 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 the existence of a collision-free trajectory respecting those priorities, we propose a provably safe trajectory planner satisfying kinodynamic constraints. The algorithm is shown to run in real time and to return safe (collision-free) trajectories. Simulation results on synthetic data illustrate the benefits of the approach.**

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 congestion motivates 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 light 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.

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 robot manipulators.

⋆ MINES ParisTech, Center for Robotics, 60 Bd St Michel 75272 Paris Cedex 06, France
† Inria Paris - Rocquencourt, RITS team, Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay, France

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], 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 problem in the coordination space into a combinatorial problem: priority assignment, and a continuous problem: finding an 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.

It is now clear that the combinatorial problem of assigning efficient priorities is inherently difficult, as noticed in [18] and developed in the priority-based framework in [12]. As a result, we will only consider in the present paper the issue of planning "good" trajectories for already assigned 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 [18], [12]. However, taking into account acceleration (and higher derivatives) constraints turns the optimization problem into a "highly non-trivial" problem (as suggested in the conclusion of the paper [18]). In the present paper, we address the challenging problem of finding safe trajectories that respect this type of constraints. In [19], 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. 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 and is proved to return collision-free trajectories respecting the assigned priorities.

Section II and III present the modelling assumptions and recall the basics of the priority-based framework of [12]. Section IV introduces the motion planner algorithm along with its safety guarantees. Finally, simulation results of Section V illustrate the efficiency of the approach.

II. MODELLING ASSUMPTIONS

A. 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], [20], [21], [22], [7] fits well intersections in a road network, where robots travel along lanes.

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 $x = (x_i)_{i \in \{1...n\}}$ and we denote $x(t)$ the evolution of x through time $t \in [0, T]$. Figure 1 illustrates the path following assumption.
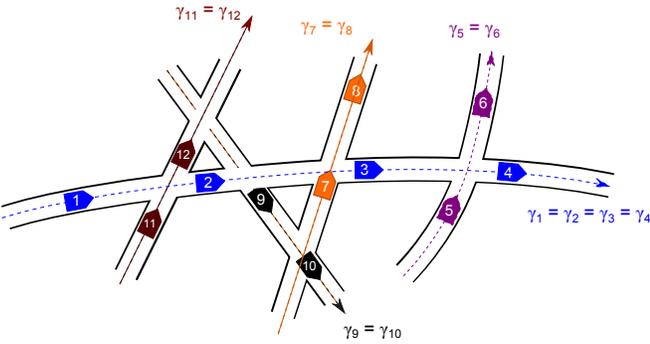

Fig. 1. The path following assumption. All robots in the same lane (depicted with the same color) travel along the same geometric path with independent velocity profiles.

The configuration space χ is known as the coordination space [23], [17], [13]. In the rest of the paper, $\{e_i\}_{1 \le i \le n}$ denotes the canonical basis of χ. 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 configurations must be excluded to avoid collisions between robots. The obstacle region χ^{obs} is the open set

of all collision configurations. $\chi^{\mathrm{free}} = \chi \backslash \chi^{\mathrm{obs}}$ denotes the obstacle-free space.

A collision occurs when two robots occupy a common region of space, so that the obstacle region can be described as the union of $n(n-1)/2$ open cylinders χ_{ij}^{obs} corresponding to as many collision pairs: $\chi^{\mathrm{obs}} = \cup_{i>j} \chi_{ij}^{\mathrm{obs}}$. Each cylinder χ_{ij}^{obs} is assumed to have an open bounded convex cross-section (in the plane generated by e_i and e_j). Figure 2 displays the obstacle region and a collision configuration for a two-path intersection. We assume $\chi^{\mathrm{obs}} \neq \emptyset$ (otherwise, coordination is not required), so the boundedness condition ensures that $\inf \chi^{\mathrm{obs}}$ and $\sup \chi^{\mathrm{obs}}$ both exist.
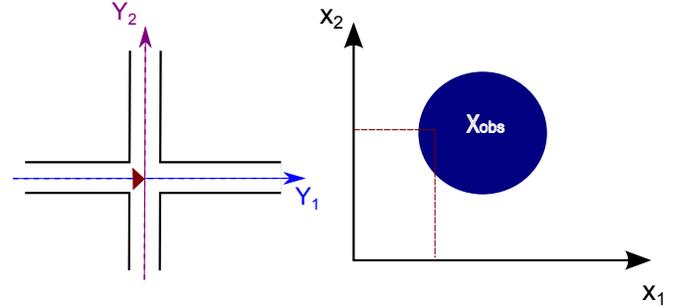

Fig. 2. The left drawing depicts two robots in collision in a 2-path-intersection. The right drawing depicts the corresponding configuration in the coordination space that belongs to the obstacle region.

B. Kinodynamic constraints

In this paper, we propose to take into account the technical constraints of the robots at the motion planning phase. These include kinematic constraints (maximum velocity, maximum curve radius, etc.) and dynamic constraints (limited acceleration, adherence, jerk, etc.). Let p denote the degree of the constraints and n the number of robots. Let $s(t) = (x\ x' \cdots x^{(p)})(t) \in \mathbb{R}^{n \times (p+1)}$ 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: $\forall i \in \{1...n\}$, $\forall t \in [0, T]$, we have $s_i(t) \in C_i$ with $C_i \subset \mathbb{R}^{p+1}$ representing the constraints for robot i and $C = \prod_{i \in \{1...n\}} C_i \subset \mathbb{R}^{n \times (p+1)}$.

Note 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 velocity can depend on the position).

We assume that the kinodynamic constraints are such that the set of reachable positions from a given state in finite time is bounded. More precisely, the set of reachable positions from state s_0 in a time-length t:

$$\chi^{\mathrm{reach}}(s_0, t) = \left\{ x(t) \middle| \begin{array}{l} x \text{ respects the constraints } C \\ s(0) = s_0 \end{array} \right\} \tag{1}$$

is assumed to be continuous with respect to s_0 and to be 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].

A. The priority graph

Consider the region $\chi^{\mathrm{obs}}_{i \succ j}$ defined as follows and depicted in Figure 3:

$$\chi^{\mathrm{obs}}_{i \succ j} = \chi^{\mathrm{obs}}_{ij} - \mathbb{R}_+ \mathbf{e}_i + \mathbb{R}_+ \mathbf{e}_j \qquad (2)$$
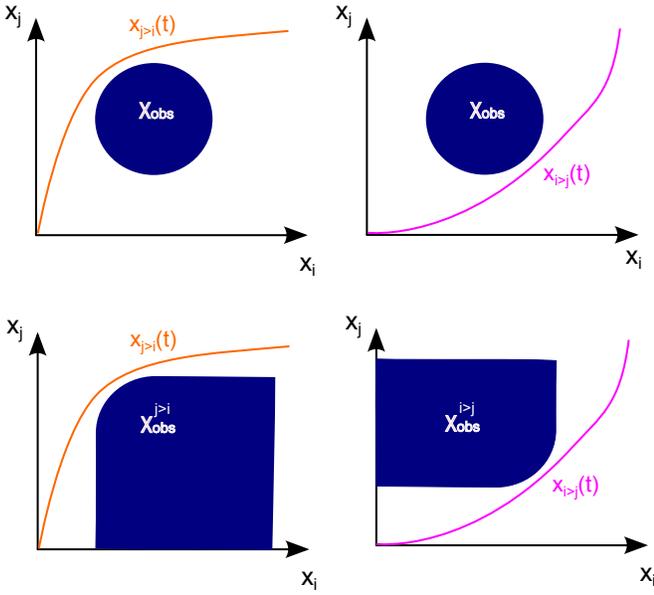
Fig. 3. The top drawings represent in the plane (x_i, x_j) the obstacle region χ^{obs}, a collision-free trajectory $x_{i \succ j}$ respecting priority $i \succ j$ and a collision-free trajectory $x_{j \succ i}$ respecting priority $j \succ i$. The bottom drawings depict $\chi^{\mathrm{obs}}_{i \succ j}$ and $\chi^{\mathrm{obs}}_{j \succ i}$.

We define a natural binary relation corresponding to priority relations between robots. A collision-free trajectory x induces a binary relation $\succ$ on the set $\{1...n\}$ as follows. For $i \neq j$ s.t. $\chi^{\mathrm{obs}}_{ij} \neq \emptyset$, $i \succ j$ if x is collision-free with $\chi^{\mathrm{obs}}_{i \succ j}$.

The priority relation can be described by a graph G with nodes $\{1...n\}$, where each edge represents the relative priority of a pair of robots. Given a collision-free trajectory x, the priority graph is defined as the oriented graph G whose vertices are $V(G) = \{1...n\}$ and such that there is an edge from i to j if $i \succ j$, we write $(i, j) \in E(G)$ where $E(G)$ denotes the edge set. An example of a priority graph for 3 robots along 3 distinct paths is described in Figure 4.

B. Problem formulation

The initial state of the robots is s^{init}, and the goal region is $\chi^{\mathrm{goal}} = \left(\sup \chi^{\mathrm{obs}} + \mathbb{R}^n_+ \right) \subset \chi^{\mathrm{free}}$. A feasible trajectory for the considered problem is a trajectory $x : [0, T] \to \chi^{\mathrm{free}}$ respecting constraints C such that $s(0) = s^{\mathrm{init}}$ and $x(T) \in$
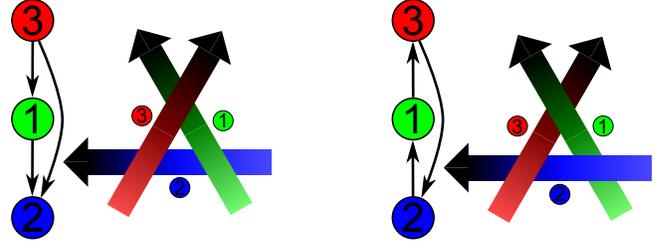

Fig. 4. Two representations of priority relations. Robots along a path in foreground have priority over robots along a path in background.

χ^{goal}. The multiple robot motion planning problem consists of finding a feasible trajectory. The benefit of the priority-based approach is that the priority graph captures the discrete part of the problem that consists of assigning the relative order of robots through the intersection.

When the priority graph G is fixed, for all $(i, j) \in E(G)$, the trajectory must be collision-free with regards to $\chi^{\mathrm{obs}}_{i \succ j}$. Given, a priority graph G, the collision region with regards to priorities G is merely defined as:

$$\chi^{\mathrm{obs}}_G = \bigcup_{(i,j) \in E(G)} \chi^{\mathrm{obs}}_{i \succ j} \qquad (3)$$

It is natural to define $\chi^{\mathrm{free}}_G = \chi \setminus \chi^{\mathrm{obs}}_G$, so that $\{\chi^{\mathrm{free}}_G, \chi^{\mathrm{obs}}_G\}$ form a partition of χ. In this paper, we focus on the problem on finding a feasible trajectory respecting assigned priorities, i.e. a trajectory $x : [0, T] \to \chi^{\mathrm{free}}_G$ respecting constraints C such that $s(0) = s^{\mathrm{init}}$ and $x(T) \in \chi^{\mathrm{goal}}$.

IV. MOTION PLANNER WITH ASSIGNED PRIORITIES

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 can not 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

The minimal (resp. maximal) trajectory from state s^0, denoted $\underline{x}(s_0, t)$ (resp. $\overline{x}(s_0, t)$), are defined bellow:

$$\begin{aligned} \underline{x}(s_0, t) &= \min \chi^{\mathrm{reach}}(s_0, t) \\ \overline{x}(s_0, t) &= \max \chi^{\mathrm{reach}}(s_0, t) \end{aligned}$$

These are the lower and upper bounds of the hypercube $\chi^{\mathrm{reach}}(s_0, t)$. 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:

$$C^{\mathrm{acc}}_i = \left\{ (x_i, x'_i, x''_i) \,\middle|\, \begin{array}{l} 0 \leq x'_i \leq v^{\mathrm{max}}_i \\ a^{\mathrm{min}}_i \leq x''_i \leq a^{\mathrm{max}}_i \end{array} \right\} \qquad (4)$$
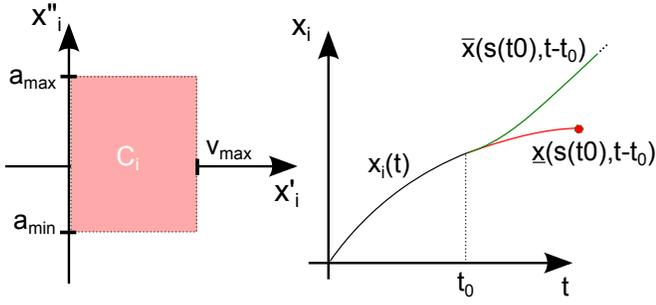
Fig. 5. The left drawing depicts an example of kinodynamic constraints where robots have uniform minimal/maximal velocity and acceleration along their paths. The right drawing depicts the corresponding minimal/maximal trajectories $\underline{x}(s(t_0), t)$ and $\overline{x}(s(t_0), t)$.

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 virtual path χ^{virtual} (line 12) that would be followed by the robots if:
  - in the next time step, robot i accelerates as much as possible while all other robots decelerate as much as possible (lines 6-11);
  - afterwards, all robots including i brake as much as possible (see the second term of the concatenation at line 12)
- If this virtual path is such that no collision and priorities are respected, it means there exists a failsafe maneuver such that robot i accelerates as much as possible in the next time step, 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 or minimal trajectory in the next time-step.

The defined trajectory thus appears as a natural extension of the left-greedy trajectory introduced in [18], in the sense that in the absence of kinodynamic constraints ($p = 1$), it coincides with it. Indeed, in this case the robots can stop instantly and the block from Line 6 to Line 17 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 from $s(t)$ is collision-free, the state $s(t)$ is not an "Inevitable Collision State" (ICS), as defined in [24] because the braking trajectory is collision-free, that is, there exists a particular collision-free trajectory starting from state $s(t)$.

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 feasible trajectory respecting priorities defined by G and the initial state s^{init} is such that the initial braking*

Algorithm 1 The motion planner with assigned priorities

Input: s^{init}, feasible priority graph G
    function MAXIMALLYAGRESSIVETRAJECTORY
        $T \leftarrow 0$
        $s(0) \leftarrow s^{\text{init}}$
        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:             $\tilde{s}_i(t) \leftarrow \overline{s}_i(s(T), t)$
              end for
              $\chi^{\text{virtual}} \leftarrow \pi(\tilde{s}([0, \Delta T])) \cup \underline{x}(\tilde{s}(\Delta T), \mathbb{R}_+)$
              if $\exists (j, i) \in E(G)$ s.t. $\chi^{\text{virtual}} \cap \chi^{\text{obs}}_{j \succ i} \neq \emptyset$ **then**
                $s_i(T + \Delta T) \leftarrow \underline{s}_i(s(T), \Delta T)$
15:            **else**
                $s_i(T + \Delta T) \leftarrow \overline{s}_i(s(T), \Delta T)$
              end if
          end for
          $T \leftarrow T + \Delta T$
20:       **end while**
        return $(x(t))_{t=0\cdots T}$
    end function

---

trajectory $\underline{x}(s^{\text{init}}, t)$ is collision-free. Then, for sufficiently small ΔT, Algorithm 1 terminates and returns a (collision-free) feasible trajectory respecting priorities G.

Proof. It is assumed that that there exists some feasible trajectory respecting priorities defined by G, so that G is a feasible priority graph as defined in [12], 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 i_0 at a coordinate lower than $\sup\{x_{i_0} : x \in \chi^{\text{obs}}\}$ 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.

Now we prove that, at every time step, the braking trajectory from the current state is collision-free. We begin with a preliminary useful property, that is a direct consequence of the definition of $\chi^{\text{obs}}_{j \succ i}$ and is easily seen on Figure 3.

Property 1. *Given $i, j \in \{1...n\}$ and two configurations $x, y \in \chi$ satisfying $y_j \geq x_j$ and $y_i \leq x_i$, we have:*

$$x \in \chi^{\text{free}}_{j \succ i} \Rightarrow y \in \chi^{\text{free}}_{j \succ i} \qquad (5)$$

The initial braking trajectory $\underline{x}(s^{\text{init}}, t)$ is assumed to be collision-free. Now, assume that for some $t_0 = k\Delta T$, $\underline{x}(s(t_0), t)$ is collision-free. In the next time step, for any priority $(j, i) \in E(G)$, there are two options for the robot with lower priority:

- either i brakes as much as possible. The fact that $\underline{x}(s(t_0), t)$ is collision-free with $\chi^{\mathrm{obs}}_{j \succ i}$ implies that $\underline{x}(s(t_0 + \Delta T), t)$ is also collision-free by Property 1, since we have for all $t \geq 0$:

$$\underline{x}_i(s(t_0 + \Delta T), t) = \underline{x}_i(s(t_0), t + \Delta T) \quad (6)$$
$$\underline{x}_j(s(t_0 + \Delta T), t) \geq \underline{x}_j(s(t_0), t + \Delta T) \quad (7)$$

- or i accelerates as much as possible; in this case, the virtual path χ^{virtual} is collision-free with respect to $\chi^{\mathrm{obs}}_{j \succ i}$. Then, consider the state $\tilde{s}(\Delta T)$ defined as:

$$\tilde{s}_j(\Delta T) = \underline{s}_j(s(t_0), \Delta T) \quad (8)$$
$$\tilde{s}_i(\Delta T) = \overline{s}_i(s(t_0), \Delta T) \quad (9)$$

Since the virtual path χ^{virtual} is collision-free with $\chi^{\mathrm{obs}}_{j \succ i}$, $\underline{x}(\tilde{s}(\Delta T), t)$ is also collision-free (see the second term in the concatenation at Line 12). It implies that $\underline{x}(s(t_0 + \Delta T), t)$ is also collision-free by Property 1, since we have for all $t \geq 0$:

$$\underline{x}_i(s(t_0 + \Delta T), t) = \underline{x}_i(\tilde{s}(\Delta T), t) \quad (10)$$
$$\underline{x}_j(s(t_0 + \Delta T), t) \geq \underline{x}_j(\tilde{s}(\Delta T), t) \quad (11)$$

Hence, at every time step, the braking trajectory is collision-free. Now, we prove that there is no collision between time steps. Again, at every time step $t_0 = k\Delta T$, the are two options:

- either i brakes as much as possible. The fact that $\underline{x}(s(t_0), t)$ is collision-free with $\chi^{\mathrm{obs}}_{j \succ i}$ implies that for $t \in [t_0, t_0 + \Delta T]$, $s(t)$ is also collision-free by Property 1, since we have for all $t \in [t_0, t_0 + \Delta T]$:

$$x_i(t) = \underline{x}_i(s(t_0), t) \quad (12)$$
$$x_j(t) \geq \underline{x}_j(s(t_0), t) \quad (13)$$

- or i accelerates as much as possible; in this case, the virtual path χ^{virtual} is collision-free with respect to $\chi^{\mathrm{obs}}_{j \succ i}$. Then, consider the trajectory $\tilde{x}(t)$ for $t \in [0, \Delta T]$ defined as:

$$\tilde{x}_j(t) = \underline{x}_j(s(t_0), t) \quad (14)$$
$$\tilde{x}_i(t) = \overline{x}_i(s(t_0), t) \quad (15)$$

Since χ^{virtual} is collision-free with $\chi^{\mathrm{obs}}_{j \succ i}$, $\tilde{x}(t)$ is also collision-free (see the first term in the concatenation at Line 12). It implies that $s(t)$ is also collision-free for $t \in [t_0, t_0 + \Delta T]$ by Property 1, since we have:

$$s_i(t) = \tilde{s}_i(t - t_0) \quad (16)$$
$$s_j(t) \geq \tilde{s}_j(t - t_0) \quad (17)$$

As a result, x is collision-free with respect to χ^{obs}_G at every time-step and reaches χ^{goal}: it is a collision-free feasible trajectory respecting priorities G. $\square$

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.

$$\forall i \in \{1...n\}, C_i^{\mathrm{acc}} = \left\{ (x_i, x_i', x_i'') \,\middle|\, \begin{array}{l} 0 \leq x_i' \leq v^{\max} \\ a^{\min} \leq x_i'' \leq a^{\max} \end{array} \right\}. \quad (18)$$

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 if it reaches the conflicting region first. This priority assignment policy can lead to deadlock configurations (see [12]), 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/bJHdf3AbIlI.
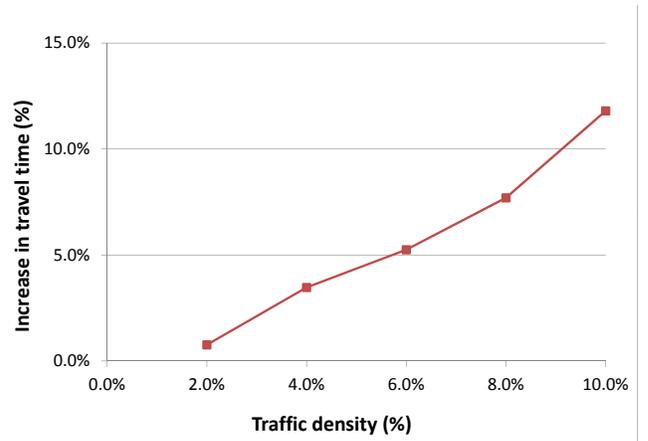

Fig. 6. Simulation results: plot of the averaged increase in travel time against the traffic density for the intersection of Figure 7
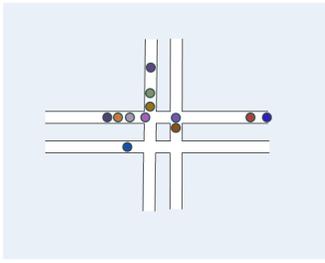
Fig. 7. The 4-path-intersection used for simulations

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 since there exists a collision-free braking maneuver at any time.

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 feasible trajectory respecting the assigned priorities. This reflects that the method proposed in this paper is inherently a feedback motion planning approach (see [17], chapter 8). We are currently turning the planning algorithms of this paper into a feedback control law that aims at coordinating robots with assigned priorities. The idea is to define a control law $g^G(s)$ that maps every state s to the control to apply in the next time step. The control law g^G is in charge of coordination, ensuring that collisions are avoided and that priorities G are respected. Robots do not have to follow precisely a planned trajectory, they just have to be aware of the priorities and to respect the control law. The benefit of the approach is that it ensures safe coordination as long as priorities G are respected, which is much easier to robustly ensure than following precisely a planned trajectory. This opens avenues to build multiple robot coordination systems much more robust with regards to uncertainty in control and sensing.

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.

[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.

[9] Ismail and Sahin, "Railway traffic control and train scheduling based oninter-train conflict management," *Transportation Research Part B: Methodological*, vol. 33, no. 7, pp. 511 – 534, 1999.

[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] 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.

[19] 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.

[20] 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.

[21] C. Hafner, Cunningham and D. Vechhio, "Automated vehicle-to-vehicle collision avoidance at intersections," 2011.

[22] 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.

[23] 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.

[24] T. Fraichard and H. Asama, "Inevitable collision states - a step towards safer robots?" *Advanced Robotics -Utrecht-*, vol. 18, no. 10, pp. 1001–1024, 2004.