Priority-based coordination of robots

Jean Gregoire, Silvère Bonnabel, Arnaud De La Fortelle

Priority-based coordination of robots

Jean Gregoire[a,*], Silvère Bonnabel[a], Arnaud de La Fortelle[a,b]

[a] *MINES ParisTech, Center for Robotics*
60, boulevard Saint Michel
75272 PARIS Cedex 06, FRANCE
Phone number: +33(0)140519408
[b] *Inria Paris - Rocquencourt, RITS team*

Abstract

This paper addresses the problem of coordinating multiple robots travelling through an intersection along fixed paths with positive velocities and kinodynamic constraints. The approach relies on a novel tool: a priority graph that encodes the relative order of the robots at the intersection. The overall planning approach can be decomposed into two key components as follows. The entry of robots into the intersection is managed by an intersection controller that assigns priorities. Within the intersection area, robots are controlled by a control law that preserves assigned priorities, avoids collisions, and is robust to unexpected decelerations of some robots occurring randomly.

Keywords: multiple robots, coordination, motion planning, control, cooperative, robustness, priority graph.

1. Introduction

We consider the problem of coordinating a collection of cooperative robots at an intersection area, motivated by applications such as coordinating a fleet of automated guided vehicles in a factory, or automated cooperative vehicles in a fully automated transportation system. Due to the promises in autonomous cars design, automated intersection management has attracted much interest recently [1, 2]. Two main goals motivate the research in this topic. The first one is to avoid accidents due to collisions that occur particularly at intersections and because of human error (the leading factor in most of road accidents). The second one is to enhance road traffic efficiency, given that intersections represent bottlenecks in the traffic network. Intelligent cooperative vehicles are expected to reduce congestion, which is one of the major problems in today's metropolitan transportation networks.

*Corresponding author
Email addresses: jean.gregoire@mines-paristech.fr (Jean Gregoire), silvere.bonnabel@mines-paristech.fr (Silvère Bonnabel), arnaud.de_la_fortelle@mines-paristech.fr (Arnaud de La Fortelle)

Preprint submitted to Systems and Control Letters *May 5, 2014*

1.1. Literature review

Multiple robot coordination with unconstrained paths is a problem of high combinatorial complexity. In [3], a path-velocity decomposition allowing to reduce the problem's complexity was first proposed. In this setting, each robot is assumed to move along a predefined path and then the velocity profiles of the robots along their assigned paths are optimized. The problem is thus decomposed into a trajectory tracking problem via a low-level controller, and a high-level planning problem, the latter leading to the realm of fixed-path coordination algorithms (see, e.g., [4]). The configuration of each robot boils down to its curvilinear position on its path and the configuration space of the whole system is then called the coordination space. It is a n-dimensional space where n denotes the number of robots going through the intersection. To prevent collisions between robots, some configurations of the coordination space must be excluded: they constitute the so-called obstacle region. The approaches based on the configuration space turn the motion planning problem into the geometric problem of searching a collision-free path for a composite robot in a n-dimensional space. The approach has since become standard in motion planning, see e.g., [5, 6, 7]. In the coordination problem, the obstacle region has a cylindrical shape [4, 5]. Reference [8] studies the problem of finding Pareto-optimal trajectories for the coordination problem. It highlights the existence of locally Pareto-optimal trajectories in each homotopy class of trajectories. However, enumerating locally optima in each homotopy class to find a globally optimal trajectory is a problem of high combinatorial complexity. That complexity led researchers to develop the so-called prioritized motion planning method.

First introduced in [9], prioritized motion planning avoids the complexity of searching a trajectory in the n-dimensional coordination space. Instead, it consists of planning the trajectory of each robot sequentially: robots for which motion has already been planned are considered as dynamic obstacles. The approach has been widely and successfully utilized since then, see e.g., [10]. Even if prioritized motion planning is not explicitly mentioned, the approach of [2, 11] for autonomous intersection management also belongs to the family of prioritized motion planning, because the trajectory of robots are planned sequentially. The collision-time approach of [12] is an extension for robots constrained to follow paths with a fixed velocity profile.

In motion planning, the planning phase and the control phase are generally decoupled, as a low-level feedback controller is assumed to track the trajectory computed by the planner. Uncertainty is thus often handled during the control phase. Feedback motion planning is a way to take into account uncertainty in an implicit way at the planning level (see [4], chapter 8). The motion plan is defined by the feedback plan that returns the action to apply given the current state of the system. The approach has been widely used for multiple robot motion planning using potential field functions from which the control vector field is derived (see, e.g. [13]). Even when global convergence to destination can be guaranteed, the weakness of the method resides in its reactive nature that can lead the system to configurations that are not efficient in terms of travel time.

Finally, the theory of multiagent systems also discusses the problem of multiagent coordination [14]. The multiple robot motion planning problem can be considered as a coordination problem where robots are collaborative agents. Individual decisions (planned trajectories) should be taken in a manner that results in a good joint utility for the fleet of robots. The key tool introduced in the theory of multi agent systems is the coordination graph. It represents, locally, the interaction between the actions of robots in the utility function. The coordination problem then boils down to solving several smaller subproblems. The approach is utilized in [15] for multiple robot distributed coordination. These methods highlight the benefit of defining a graph to represent the interaction between robots and defining relative orders. The relevance of coordination graphs combined with the cylindrical structure of the coordination space led us to develop the priority-based framework exploited in the present paper.

1.2. Contributions and organization

In the present paper, we aim at building a multiple robot coordination system at an intersection area that yields efficient trajectories in terms of travel time and is robust with respect to large deviations due to unexpected events. Prioritized motion planning methods build collision-free and efficient trajectories respecting kinodynamic constraints, but if there are large deviations from the planned trajectories, safety cannot be guaranteed any more: these methods require precise trajectory tracking control. We propose to use a prioritized motion planning approach only to plan an efficient relative order of the robots through the intersection. However, the precise trajectory is not planned in advance. A control law (mapping the current state to a control action) ensures collision avoidance and the preservation of the planned relative order: this is a feedback motion planning approach. It relies on a novel tool: a priority graph that encodes the relative order of the robots. The overall planning approach can be decomposed into two key components as follows. The entry of robots into the intersection is managed by an intersection controller that assigns priorities. Within the intersection area, robots are controlled by a control law that preserves assigned priorities, avoids collisions, and is robust to unexpected decelerations of some robots occurring randomly.

In Section 2, we recall the priority-based framework introduced in the preliminary conference paper [16]. The benefit of the approach is that it decomposes the coordination problem into a discrete problem (priority assignment, i.e. choosing a priority graph), and a continuous problem (controlling robots with assigned priorities). A solution to the continuous problem is provided in Section 3 for robots whose acceleration can be controlled. The control law ensures collision avoidance and preserves assigned priorities. Moreover, the state of the robots remains brake safe, which guarantees that at any point of time, a robot may brake independently from other robots' decisions without causing any collision. This latter fact is proved using the theory of monotone systems [17], and makes the coordination system robust to a large class of perturbations. The proposed control law can be viewed as an extension of the two-vehicle coordination system proposed in [18], made possible by the introduction of the priority graph. In Section 4, we propose an overall coordination system which priority assignment policy is inspired from the prioritized motion planning approaches. Robots are required to request an intersection controller the right to enter a control area. The intersection controller assigns priorities. Once accepted into the control area, robots are controlled by the control law of Section 3. To the authors' knowledge, this is the first multiple robot coordination system combining a global motion planning approach (for priority assignment) with a feedback control law (for control under assigned priorities). The theoretical benefits in terms of robustness are illustrated by qualitative simulation results presented in Section 5.

2. The priority-based framework

Consider the problem of coordinating the motion of a collection of robots R in a two-dimensional space. Every robot $i \in R$ follows a particular path $\gamma_i \subset \mathbb{R}^2$ and we let $x_i \in \mathbb{R}$ denote its curvilinear coordinate along the path (see Figure 1). $x := (x_i)_{i \in R}$ indicates the configuration of all robots. $x \in \chi := \mathbb{R}^n$ where n denotes the number of robots going through the intersection. The configuration space χ is known as the coordination space [19]. In the rest of the paper, $\{\mathbf{e}_i\}_{1 \leq i \leq n}$ denotes the canonical basis of χ.

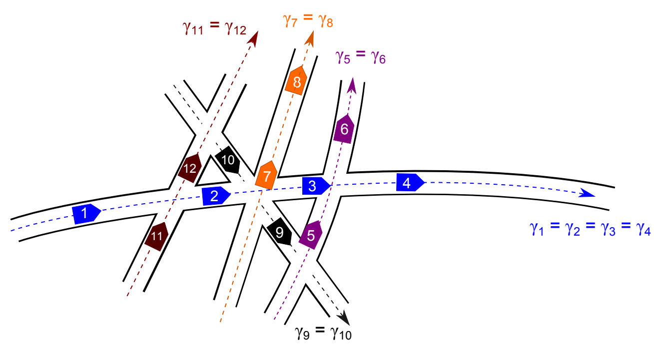

Figure 1: The fixed paths assumption. Every robot travels along an assigned path.

Some configurations must be excluded to avoid collisions between robots (see Figure 2). The obstacle region χ^{obs} is the open set of all collision configurations. $\chi^{\text{free}} := \chi \backslash \chi^{\text{obs}}$ denotes the obstacle-free space. Letting χ_{ij}^{obs} denote the set of configurations where i and j collide, it is easily seen that χ_{ij}^{obs} is an (open) cylinder, and the obstacle region merely appears as the union of $n(n-1)/2$ cylinders, that is, $\chi^{\text{obs}} = \cup_{\{i,j\}} \chi_{ij}^{\text{obs}}$ [4]. Every cylinder χ_{ij}^{obs} is assumed to have an open bounded convex cross-section (in the plane generated by $\mathbf{e}_i$ and $\mathbf{e}_j$). The boundedness condition on χ^{obs} is rather technical but ensures that the whole intersection lies in a bounded region: the intersection area. We assume

$\chi^{\text{obs}} \neq \emptyset$ (otherwise, coordination is not required), so the boundedness condition ensures that $\inf \chi^{\text{obs}}$ and $\sup \chi^{\text{obs}}$ both exist.

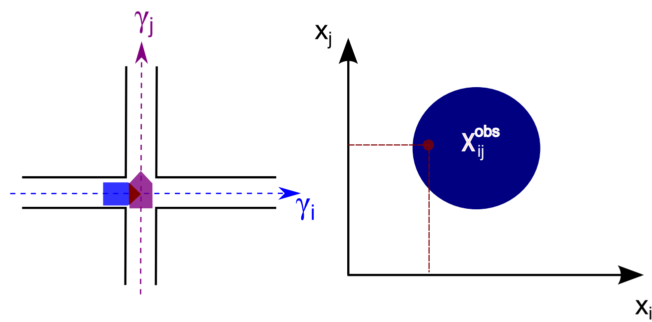

Figure 2: The left drawing depicts two paths with two robots in collision in the current configuration. The left drawing shows the obstacle region associated to the two paths in the coordination space and the collision configuration $(x_i, x_j) \in \chi^{\text{obs}}_{ij}$ corresponding to the collision of the left drawing.

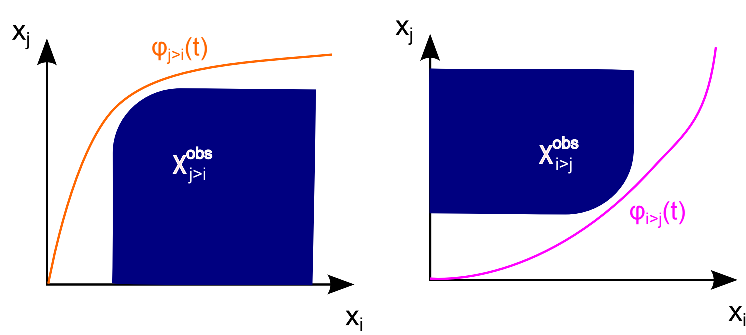

Figure 3: The collision cylinders with assigned priorities $\chi^{\text{obs}}_{i \succ j}$ and $\chi^{\text{obs}}_{j \succ i}$.

In this paper, we will only consider motions where all robots have a non-negative velocity at all times: non-positive velocities would result in robots that move backwards in the intersection area. Paths in the coordination space satisfying the non-negative velocity assumptions, starting from an initial configuration lower than $\inf \chi^{\text{obs}}$ and ending at a final configuration greater than $\sup \chi^{\text{obs}}$, will be called feasible. Because of the non-negative velocity constraint, for every couple of robots with a non-empty collision region, one of the robots necessarily passes "before" or "after" the other one. In the coordination space, a feasible path passes below or above the collision cylinder as depicted in Figure 3 where $\varphi(t)$ denotes a feasible motion in the coordination space. This reflects the intuitive notion of priority. Let $\chi^{\text{obs}}_{i \succ j} \subset \chi$ denote the set defined below:

$$\chi^{\text{obs}}_{i \succ j} := \chi^{\text{obs}}_{ij} - \mathbb{R}_+ \mathbf{e}_i + \mathbb{R}_+ \mathbf{e}_j \tag{1}$$

Figure 3 illustrates the sets $\chi^{\text{obs}}_{i \succ j}$ and $\chi^{\text{obs}}_{j \succ i}$. The geometry of the coordination space thus leads us to define a natural binary relation corresponding to priority relations between robots: a very familiar and intuitive notion in real life. Any feasible path induces a binary relation $\succ$ on the set R as follows. For $i \neq j$ s.t. $\chi_{ij}^{\text{obs}} \neq \emptyset$, $i \succ j$ if the path is collision-free with $\chi_{i \succ j}^{\text{obs}}$: we say robot i has priority over robot j. We let the priority graph be the oriented graph G whose vertices are $V(G) := R$ 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. When priorities are assigned, for each priority $i \succ j$, the fixed-priority collision cylinder $\chi_{i \succ j}^{\text{obs}}$ must be avoided. Given a priority graph G, the collision region with regards to priorities defined by G is merely defined as:

$$\chi_G^{\text{obs}} := \bigcup_{(i,j) \in E(G)} \chi_{i \succ j}^{\text{obs}} \tag{2}$$

It is natural to define $\chi_{i \succ j}^{\text{free}} := \chi \setminus \chi_{i \succ j}^{\text{obs}}$ and $\chi_G^{\text{free}} := \chi \setminus \chi_G^{\text{obs}}$, so that $\{\chi_G^{\text{obs}}, \chi_G^{\text{free}}\}$ form a partition of χ. We say a feasible path respects priorities G if it takes values in χ_G^{free}.

3. A priority-preserving control law

In this section, we assume the acceleration of the robots can be controlled, and we propose a (centralized) control law aimed at coordinating multiple robots with assigned priorities and kinodynamic constraints in the intersection area. The method is inspired by the works [18, 20, 21] for the coordination of two vehicles (the priority graph being then reduced to one oriented edge).

3.1. The multiple robot system as a monotone controlled system

Each robot i is modelled as a second-order controlled system with state $s_i = (x_i, v_i) \in S_i := \mathbb{R} \times [0, \overline{v}_i]$, whose evolution is described by the differential equation:

$$\dot{x_i}(t) = v_i(t) \tag{3}$$

$$\dot{v_i}(t) = \mathbf{u}_i(t)\ \delta(\mathbf{u}_i(t), v_i(t)) \tag{4}$$

where $\mathbf{u}_i : \mathbb{R}_+ \to U_i$ is the control of robot i and $\overline{v}_i$ denotes the non-negative speed limit for robot i. We let $U_i := [\underline{u}_i, \overline{u}_i]$ be the set of feasible control values. $\underline{u}_i < 0$ represents the maximum brake control value and $\overline{u}_i > 0$ represents the maximum throttle control value. δ is a binary function merely ensuring that $v_i \in [0, \overline{v}_i]$ at all times, that is, $\delta(\mathbf{u}_i(t), v_i(t)) = 1$ except for $v_i(t) = 0$ and $\mathbf{u}_i(t) < 0$, and for $v_i(t) = \overline{v}_i$ and $\mathbf{u}_i(t) > 0$, where it vanishes.

The control is assumed to be updated in discrete time every $\Delta T > 0$:

$$\forall k \in \mathbb{N}, \forall t \in [k\Delta T, (k+1)\Delta T), \mathbf{u}_i(t) \equiv \mathbf{u}_i(k\Delta T) \tag{5}$$

The time interval $[k\Delta T, (k+1)\Delta T)$ will be referred to as (time) slot k. For the sake of simplicity we let $\Delta T := 1$ in the sequel. We let $\mathbf{U}_i$ denote the set of controls $\mathbf{u}_i : \mathbb{R}_+ \to U_i$ piecewise constant on intervals $[k, k+1)$, $k \in \mathbb{N}$. We let $t \mapsto \Phi_i(t, s_i, \mathbf{u}_i)$ denote the flow of the system starting at initial condition $s_i \in S_i$ with control $\mathbf{u}_i \in \mathbf{U}_i$.

We also define the vectorial state $s := (s_i)_{i \in R} \in S$, the vectorial control $\mathbf{u} := (\mathbf{u}_i)_{i \in R} \in \mathbf{U} := \prod_{i \in R} \mathbf{U}_i$, and the vectorial flow: $\Phi(t, s, \mathbf{u}) := (\Phi_i(t, s_i, \mathbf{u}_i))_{i \in R}$. We let $\underline{u} := (\underline{u}_i)_{i \in R}$, $\overline{u} := (\overline{u}_i)_{i \in R}$ and we define the constant controls $\underline{\mathbf{u}}(t) := \underline{u}$ and $\overline{\mathbf{u}}(t) := \overline{u}$. We introduce partial orders as follows:

$$\forall \mathbf{u}_i^1, \mathbf{u}_i^2 \in \mathbf{U}_i, \mathbf{u}_i^1 \preceq \mathbf{u}_i^2 \quad \text{if} \quad \forall t \geq 0, \mathbf{u}_i^1(t) \leq \mathbf{u}_i^2(t) \tag{6}$$

$$\forall s_i^1 = (x_i^1, v_i^1), s_i^2 = (x_i^2, v_i^2) \in S_i, s_i^1 \preceq s_i^2 \quad \text{if} \quad x_i^1 \leq x_i^2 \text{ and } v_i^1 \leq v_i^2 \tag{7}$$

$$\forall \Phi^1, \Phi^2 : \mathbb{R}_+ \to S, \Phi^1 \preceq \Phi^2 \quad \text{if} \quad \forall t \geq 0, \Phi^1(t) \preceq \Phi^2(t) \tag{8}$$

The controlled system (3)-(4) is a monotone control system [17] with regards to the relative orders defined above. More precisely, the following key property holds:

Property 1 (Order preservation)**.** *The flow $t \mapsto \Phi_i(t, s_i, \boldsymbol{u}_i)$ is order-preserving with regards to s_i and $\boldsymbol{u}_i$.*

Note that in our open loop model, control $\mathbf{u}_i$ only acts on robot i, that is, $\mathbf{u}$ is a collection of independent controls: it does not achieve any kind of coordination between the robots. The control law introduced in the sequel is precisely aiming at coordinating the robots to avoid collisions and respect priorities.

3.2. The proposed control law

We define projection operators as follows: $\pi_x(s) := x$ and $\pi_{x,i}(s) := \pi_{x,i}(s_i) := x_i$. G denotes a given priority graph. Define the set of brake safe states as follows:

$$B_G := \{s \in S : \pi_x(\Phi(\mathbb{R}_+, s, \underline{\mathbf{u}})) \subset \chi_G^{\text{free}}\} \subset S \tag{9}$$

According to the above definition, a state $s \in S$ is brake safe if, starting at initial condition s under maximum brake control, the system remains in χ_G^{free}. In particular, a state $(x, 0)$ with $x \in \chi_G^{\text{free}}$ is brake safe, so B_G is not empty provided χ_G^{free} is not empty. Brake safety is more conservative than remaining in the escape set proposed in [18], which includes all states from which there exists at least one control (not necessarily $\underline{\mathbf{u}}$) avoiding future collisions. It is also more conservative than not entering an inevitable collision state as defined in [22] where neither the geometric path in $\mathbb{R}_2$ nor the control to avoid collisions are fixed. In the context of multiple robot coordination, checking whether a state is an inevitable collision state, and computing the escape set are of high computational complexity [20]. In contrast, verifying that a state is brake safe consists of computing a finite time single flow and checking if collisions occur for all pairs of robots, yielding a quadratic complexity. Moreover, constraining the state to be brake safe at all times comes with robustness properties, as at any time, all robots can be stopped with maximum brake control without colliding.

Now, we propose to build a control law $g^G : S \to U$ such that starting from an initial brake safe state in B_G, the flow of the system controlled by the control law g^G is ensured to remain in B_G (thus being collision-free and respecting priorities G). In other words, using the terminology of [23], B_G shall be positively invariant for the system under control law g^G. The liveness property of the control law, i.e. its ability to let each robot reach its goal will be studied in Section 4 and we first focus on collision avoidance.

The rationale for our control law is as follows. Consider a robot i and a robot j that has priority over i. Given an initial configuration of the two robots, the worst-case scenario is when j brakes whereas i accelerates in the next time slot. If the trajectory of the system in the next time slot under that worst-case scenario is collision-free and if the reached state is brake safe, robot i may accelerate in any case. Otherwise, it is required to brake. This is formalized below.

Let $\mathbf{u}_i^{\text{impulse}} \in \mathbf{U}_i$ denote the impulse control for robot i defined by:

$$\mathbf{u}_i^{\text{impulse}}(k) := \begin{cases} \overline{u}_i & \text{if } k = 0 \\ \underline{u}_i & \text{if } k \geq 1 \end{cases} \tag{10}$$

Now let $\tilde{\mathbf{u}}^i$ denote the worst-case vectorial control with regards to i defined componentwise by:

$$\tilde{\mathbf{u}}_j^i := \begin{cases} \mathbf{u}_i^{\text{impulse}} & \text{if } j = i \\ \underline{\mathbf{u}}_j & \text{if } j \neq i \end{cases} \tag{11}$$

The control law can then be formulated synthetically:

$$g_i^G(s) := \begin{cases} \underline{u}_i & \text{if } \exists (j,i) \in E(G) \text{ s.t. } \pi_x(\Phi(\mathbb{R}_+, s, \tilde{\mathbf{u}}^i)) \cap \chi_{j \succ i}^{\text{obs}} \neq \emptyset \\ \overline{u}_i & \text{else.} \end{cases} \tag{12}$$

This simply means that robot i applies maximum throttle command unless the worst-case flow $t \mapsto \Phi(t, s, \tilde{\mathbf{u}}^i)$ intersects χ_G^{obs} at some point of time $t \geq 0$, in which case it applies maximum brake command. Now, in order to present our first main result, we need to introduce the following notation. Given a feedback control law $h : S \to U$, with a slight abuse of notation we let $t \mapsto \Phi(t, s, h)$ denote the vectorial flow of the system starting at initial condition $s \in S$ and controlled by $\mathbf{u} \in \mathbf{U}$ satisfying:

$$\forall k \in \mathbb{N}, \mathbf{u}(k) \equiv h(\Phi(k, s, \mathbf{u})) \tag{13}$$

Theorem 1 (Control law safety)**.** *The set of brake safe states B_G is positively invariant (in discrete time) for the system under control law g^G, i.e.:*

$$\forall s \in B_G, \forall k \in \mathbb{N}, \Phi(k, s, g^G) \in B_G \tag{14}$$

Moreover, the configuration of the system remains in χ_G^{free} through time, i.e.:

$$\forall s \in B_G, \forall t \geq 0, \pi_x(\Phi(t, s, g^G)) \in \chi_G^{\text{free}} \tag{15}$$

The above theorem asserts that under control law g^G, provided the system starts in a brake safe state, the sequence of future states at the beginning of each time slot is a sequence of brake safe states (see Equation (14)). Moreover, the flow of the system remains in χ_G^{free} in continuous time (see Equation (15)), i.e. no collision occurs and priorities are preserved. It is a direct consequence of Theorem 2 and appears as a limiting case.

3.3. Robustness issues

The control law g_i^G returns the maximum control value that robot i can safely apply, but it is in fact always safe to apply a lower control value, including letting all vehicles brake as much as possible, i.e. leading to an emergency stop. This property stated in Theorem 2 below is very valuable because for applications in intelligent transportation systems, even without considering extreme situations such as emergency stops, it is very usual that a vehicle needs to brake because of an unpredictable event such as a pedestrian crossing the road, or delay or even loss of communication.

Theorem 2 (A broad class of priority-preserving controls)**.** *Given an initial condition $s \in B_G$, and a control $\boldsymbol{u} \in \mathbf{U}$ that satisfies:*

$$\forall k \in \mathbb{N}, \boldsymbol{u}(k) \leq g^G(\Phi(k, s, \boldsymbol{u})) \tag{16}$$

The set of brake safe states B_G is positively invariant (in discrete time), i.e.:

$$\forall k \in \mathbb{N}, \Phi(k, s, \boldsymbol{u}) \in B_G \tag{17}$$

Moreover, the configuration of the system remains in χ_G^{free} through time, i.e.:

$$\forall t \geq 0, \pi_x(\Phi(t, s, \boldsymbol{u})) \in \chi_G^{\text{free}} \tag{18}$$

Proof. By induction, it is sufficient to prove that given an initial condition $s \in B_G$, the flow is collision-free for $t \in [0, 1]$ and the reached state $\Phi(1, s, \mathbf{u})$ is brake safe. We begin with a preliminary useful property, that is a direct consequence of the definition of $\chi_{j \succ i}^{\text{obs}}$ in Equation (1) and is easily seen on Figure 3.

Property 2. *Given $i, j \in R$ and two configurations $x, y \in \chi$ satisfying $y_j \geq x_j$ and $y_i \leq x_i$, we have:*

$$x \in \chi_{j \succ i}^{\text{free}} \Rightarrow y \in \chi_{j \succ i}^{\text{free}} \tag{19}$$

Now, we prove that the flow of Theorem 2 does not intersect χ_G^{obs} for $t \in [0, 1]$. Take arbitrary $t \in [0, 1]$: we have to prove that for all $(j, i) \in E(G)$, $\pi_x(\Phi(t, s, \mathbf{u})) \in \chi_{j \succ i}^{\text{free}}$. By construction of g^G, for each robot i, there are two cases:

- $g_i^G(s) = \underline{u}_i$: in this case,

$$\Phi_i(t, s, \mathbf{u}) = \Phi_i(t, s, \underline{\mathbf{u}}) \tag{20}$$

and by order-preservation, for all robots j such that $(j, i) \in E(G)$ we have:

$$\Phi_j(t, s, \mathbf{u}) \geq \Phi_j(t, s, \underline{\mathbf{u}}) \tag{21}$$

Since s is brake safe, $\pi_x(\Phi(t, s, \underline{\mathbf{u}})) \in \chi_{j \succ i}^{\text{free}}$. Hence, by Property 2, Equations (20) and (21) ensure that $\pi_x(\Phi(t, s, \mathbf{u})) \in \chi_{j \succ i}^{\text{free}}$ as well.

- $g_i^G(s) = \overline{u}_i$: by construction of the control law, $\pi_x(\Phi(t, s, \tilde{\mathbf{u}}^i)) \in \chi_G^{\text{free}}$. By order-preservation, using $\tilde{\mathbf{u}}_i^i(0) = \overline{u}_i$, we obtain:

$$\Phi_i(t, s, \tilde{\mathbf{u}}^i) = \Phi_i(t, s, \overline{\mathbf{u}}) \geq \Phi_i(t, s, \mathbf{u}) \tag{22}$$

For all robots j such that $(j, i) \in E(G)$, using $\tilde{\mathbf{u}}_j^i(0) = \underline{u}_j$, we have:

$$\Phi_j(t, s, \tilde{\mathbf{u}}^i) = \Phi_j(t, s, \underline{\mathbf{u}}) \leq \Phi_j(t, s, \mathbf{u}) \tag{23}$$

Since $\pi_x(\Phi(t, s, \tilde{\mathbf{u}}^i)) \in \chi_G^{\text{free}}$, $\pi_x(\Phi(t, s, \tilde{\mathbf{u}}^i)) \in \chi_{j \succ i}^{\text{free}}$, and by Property 2, Equations (22) and (23) ensure that $\pi_x(\Phi(t, s, \mathbf{u})) \in \chi_{j \succ i}^{\text{free}}$ as well.

As a final step, let us prove that the reached state $s^1 := \Phi(1, s, \mathbf{u})$ is brake safe. Take arbitrary $t \geq 0$: we have to prove that for all $(j, i) \in E(G)$, $\pi_x(\Phi(t, s^1, \underline{\mathbf{u}})) \in \chi_{j \succ i}^{\text{free}}$. As previously, there are two cases:

- $g_i^G(s) = \underline{u}_i$: then, $s_i^1 = \Phi_i(1, s, \underline{\mathbf{u}})$ and we have:

$$\Phi_i(t, s^1, \underline{\mathbf{u}}) = \Phi_i(1 + t, s, \underline{\mathbf{u}}) \tag{24}$$

Moreover, by order-preservation, for all j such that $(j, i) \in E(G)$: $s_j^1 \geq \Phi_j(1, s, \underline{\mathbf{u}})$. As a result, by order-preservation:

$$\Phi_j(t, s^1, \underline{\mathbf{u}}) \geq \Phi_j(1 + t, s, \underline{\mathbf{u}}) \tag{25}$$

Since s is brake safe, $\pi_x(\Phi(1 + t, s, \underline{\mathbf{u}})) \in \chi_{j \succ i}^{\text{free}}$. Hence, by Property 2, Equations (24) and (25) ensure that $\pi_x(\Phi(t, s^1, \underline{\mathbf{u}})) \in \chi_{j \succ i}^{\text{free}}$ as well.

- $g_i^G(s) = \overline{u}_i$: then, by construction of the control law, $\pi_x(\Phi(1 + t, s, \tilde{\mathbf{u}}^i)) \in \chi_G^{\text{free}}$. Define $\tilde{s}^1 := \Phi(1, s, \tilde{\mathbf{u}}^i)$. We have $\tilde{\mathbf{u}}^i(1 + \tau) = \underline{u}$ for $\tau \geq 0$. As a result, $\Phi(1 + t, s, \tilde{\mathbf{u}}^i) = \Phi(t, \tilde{s}^1, \underline{\mathbf{u}})$. Since $\pi_x(\Phi(1 + t, s, \tilde{\mathbf{u}}^i)) \in \chi_G^{\text{free}}$, $\pi_x(\Phi(t, \tilde{s}^1, \underline{\mathbf{u}})) \in \chi_G^{\text{free}}$.

  By order-preservation, using $\tilde{\mathbf{u}}_i^i(0) = \overline{u}_i$, we obtain:

$$\tilde{s}_i^1 = \Phi_i(1, s, \tilde{\mathbf{u}}^i) = \Phi_i(1, s, \overline{\mathbf{u}}) \geq \Phi_i(1, s, \mathbf{u}) = s_i^1 \tag{26}$$

For all robots j such that $(j, i) \in E(G)$, using $\tilde{\mathbf{u}}_j^i(0) = \underline{u}_j$, we have:

$$\tilde{s}_j^1 = \Phi_j(1, s, \tilde{\mathbf{u}}^i) = \Phi_j(1, s, \underline{\mathbf{u}}) \leq \Phi_j(1, s, \mathbf{u}) = s_j^1 \tag{27}$$

Hence, by order-preservation, Equations (26) and (27) imply:

$$\begin{aligned} \Phi_i(t, \tilde{s}^1, \underline{\mathbf{u}}) &\geq \Phi_i(t, s^1, \underline{\mathbf{u}}) && (28) \\ \Phi_j(t, \tilde{s}^1, \underline{\mathbf{u}}) &\leq \Phi_j(t, s^1, \underline{\mathbf{u}}) && (29) \end{aligned}$$

Since $\pi_x(\Phi(t, \tilde{s}^1, \underline{\mathbf{u}})) \in \chi_G^{\text{free}}$, $\pi_x(\Phi(t, \tilde{s}^1, \underline{\mathbf{u}})) \in \chi_{j \succ i}^{\text{free}}$, and by Property 2, Equations (28) and (29) ensure that $\pi_x(\Phi(t, s^1, \underline{\mathbf{u}})) \in \chi_{j \succ i}^{\text{free}}$ as well.

□

To illustrate the interest of Theorem 2, given priorities G and an initial condition $s \in B_G$ consider the two examples below.

Example 1 (Individual brake application)**.** Consider a control $\mathbf{u} \in \mathbf{U}$ satisfying:

$$\begin{aligned} \forall k \in \mathbb{N}, \mathbf{u}_i(k) &= \begin{cases} \underline{u}_i & \text{if } k \in K \\ g_i^G(\Phi(k, s, \mathbf{u})) & \text{else.} \end{cases} && (30) \\ \forall j \in R, j \neq i, \mathbf{u}_j(k) &= g_j^G(\Phi(k, s, \mathbf{u})) && (31) \end{aligned}$$

$i \in R$ is a particular robot and $K \subset \mathbb{N}$ is a subset of slots. Under the control described above, the system is perfectly controlled by the control law, except during slots K where the particular robot i brakes while other robots are still perfectly controlled by the control law. Such a scenario may arise, for instance, in case of a momentary communication/sensing failure for one robot: if the current state is not available, the control law cannot be applied, and a brake manoeuvre is performed instead. The condition of Theorem 2 is clearly respected since for $j \neq i$, $\mathbf{u}_j(k) = g_j^G(\Phi(k, s, \mathbf{u})) \leq g_j^G(\Phi(k, s, \mathbf{u}))$, and $\mathbf{u}_i(k) = g_i^G(\Phi(k, s, \mathbf{u})) \leq g_i^G(\Phi(k, s, \mathbf{u}))$ or $\mathbf{u}_i(k) = \underline{u}_i \leq g_i^G(\Phi(k, s, \mathbf{u}))$. Hence, the flow $t \mapsto \Phi(t, s, \mathbf{u})$ is collision-free and preserves priorities G. This illustrates that the control law is robust with regards to an individual brake application of a particular robot for an arbitrary long time, yielding a deviated but still collision-free flow respecting the assigned priorities.

Example 2 (Simultaneous brake application)**.** Consider a control $\mathbf{u} \in \mathbf{U}$ satisfying:

$$\forall k \in \mathbb{N}, \mathbf{u}(k) = \begin{cases} \underline{u} & \text{if } k \in K \\ g^G(\Phi(k, s, \mathbf{u})) & \text{else.} \end{cases} \tag{32}$$

Again, $K \subset \mathbb{N}$ is a subset of slots. Under the control described above, the system is perfectly controlled by the control law, except during slots K where all robots brake simultaneously. It may arise in case of a global failure requiring an emergency brake to be performed. Again, the condition of Theorem 2 is clearly respected since $\mathbf{u}(k) = g^G(\Phi(k, s, \mathbf{u})) \leq g^G(\Phi(k, s, \mathbf{u}))$ or $\mathbf{u}(k) = \underline{u} \leq g^G(\Phi(k, s, \mathbf{u}))$. It illustrates that the control law is robust with regards to a simultaneous brake application of all robots for an arbitrary long time, yielding again a deviated but still collision-free flow respecting the assigned priorities.

4. Overall coordination system

Given a priority graph, we know now how to coordinate the robots without violating the priorities encoded by the graph, and without colliding. However, finding a sensible graph is in fact far from trivial. Many graphs can be infeasible, that is, there is no trajectory respecting the priorities defined by it (that is, the so-called liveness property is not satisfied). Moreover, even in the case it is feasible, some unwise priorities can result in a very inefficient coordination (think of a line of robots in front an empty intersection waiting some far away robot to pass first). There are $2^{n(n-1)/2}$ possible graphs, and enumerating them all soon become numerically prohibitive as the number of robots in the intersection grows [8]. As a result, the present section proposes a simple yet sensible way to assign the priorities, inspired by the work [2], and proves that the obtained graphs are feasible. Because of the inherent complexity of the problem, and to the fact that the intersection is open, meaning the number of robots keep varying, the proposed overall coordination system proposed in Subsections 4.2 and 4.3 has a more engineering flavour than what has been done so far in the present paper.

4.1. Sufficient condition for liveness: acyclic priorities

In Section 3 we have focused on collision avoidance by building a control law under which B_G is positively invariant. Another key property in motion planning is liveness, i.e. the guarantee that every robot eventually reaches its goal. In the particular case of the problem studied here, every robot is expected to exit the obstacle region. Since it is bounded, liveness is guaranteed if every robot $i \in R$ eventually reaches position $\sup_{x \in \chi^{\text{obs}}} x_i$. This can only happens if there are no deadlocks, i.e. situations where all robots are blocked in the intersection and can not move anymore. Our preliminary work already noticed the role of cycles in deadlocks [16]. The following theorem proves that when the graph G is acyclic, and under the proposed control law g^G, liveness is guaranteed.

Theorem 3 (Sufficient condition for liveness)**.** *Given an acyclic priority graph G and an initial condition $s \in B_G$, robots eventually exit the intersection under control law g^G, i.e. the flow under control law g^G satisfies:*

$$\exists T > 0 : \forall i \in R, \pi_{x,i}(\Phi(T, s, g^G)) > \sup_{x \in \chi^{\text{obs}}} x_i \tag{33}$$

Proof. Consider the trajectory of the robots under control law g^G. G being acyclic, there exists an extremal vertex $i_1 \in R$ such that for all $j \in R$, $(j, i_1) \notin E(G)$. As a result, under the control law g^G, robot i_1 will always accelerate as much as possible and it will exit the intersection in finite time T_1.

Now, assume that at time T_m, robots $i_1 \cdots i_m$ have exited the intersection and $m < n$ (there remain some robots). G being acyclic, there exists an extremal element for the remaining robots denoted $i_{m+1} \in R \backslash \{i_1 \cdots i_m\}$ such that for all

$j \in R \setminus \{i_1 \cdots i_m\}$, $(j, i_{m+1}) \notin E(G)$. Collisions occurring only with non exited robots, for $t \geq T_m$ j will always accelerate and it will exit the intersection in finite time at instant $T_{m+1} \geq T_m$.

Iterating this process for $m = 1 \cdots n-1$ yields a sequence $(T_1 \cdots T_n)$ and all robots have exited the intersection at time $T := T_n$. $\square$

4.2. A simple priority assignment policy

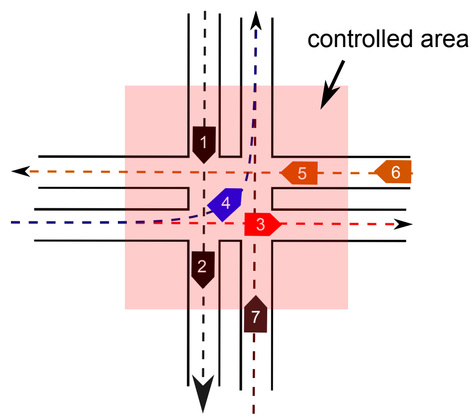

Figure 4: The control area.

We define the control area as a subset of the two-dimensional real space in which the obstacle region wholly resides (see Figure 4). For each path, an entry position and an exit position are defined. Let x_i^{entry} denote the entry position for robot i and x_i^{exit} its exit position. Robot i is in the control area if $x_i \in [x_i^{\text{entry}}, x_i^{\text{exit}}]$. Every robot is considered as an agent, and a control area controller, called the intersection controller, manages the intersection. It updates the set R of robots accepted in the control area and the priority graph G. Time is slotted, and at every time slot, the current state s of the system is available to the intersection controller and to all robots.

The entry of the control area is managed using requests sent by robots to the intersection controller. Each robot i that has not already been accepted in the control area checks at every time slot whether accelerating (or maintaining maximum velocity) during the next time slot will inevitably result in an entry into the control area. If this is the case, entry must be requested to the intersection controller, and if the entry is denied, robot i must brake. To formulate this mathematically, we compute the final (and maximal) position reached by robot i with initial state s_i under impulse control defined in Equation (10):

$$x_i^{\text{stop}}(s_i) := \max \pi_{x,i}(\Phi_i(\mathbb{R}_+, s_i, \mathbf{u}_i^{\text{impulse}})) \tag{34}$$

The condition to request entry then simply becomes $x_i^{\text{stop}}(s_i) > x_i^{\text{entry}}$. Once requests are received, the intersection controller chooses an order and processes the requests one by one. The idea is to spend as little time as possible in the intersection area, as noticed in [2]. Thus a robot is accepted into the control area only if it can travel with maximum throttle command and with lowest priority. The second point is key: assigning every accepted robot the lowest priority with regards to robots already accepted into the control area leads to a necessarily acyclic graph, enforcing liveness (see Theorem 3). This can be formulated as follows. Consider a robot i that requests the entry of the control area. To decide to accept it or not, a virtual trajectory is built that consists of applying control $\overline{u}_i$ constantly to robot i while robots $j \neq i$ follow the trajectory that they would have followed in the absence of i. Let $\tilde{s}(\tau)$ denote this virtual trajectory, for all $\tau \geq 0$:

$$\tilde{s}_i(\tau) := \Phi_i(\tau, s_i, \overline{\mathbf{u}}_i) \tag{35}$$

$$\forall j \in R, \tilde{s}_j(\tau) := \Phi_j(\tau, s, g^G) \tag{36}$$

Define the virtual priority graph that consists of adding to G edges (j, i) for all robots $j \in R$. If all the states travelled along the virtual trajectory are brake safe with regards to the virtual priority graph, the request is accepted. Otherwise, the request is rejected. When the request is accepted, the priority graph is updated, and the newly accepted robot is assigned the lowest priority (the virtual priority graph becomes the current priority graph). Moreover, the robot is added to the set of robots accepted into the control area. Note that the described algorithm ensures the priority relation to be a relative order, that is G to be a directed acyclic graph at all times.

4.3. Proposed overall coordination system

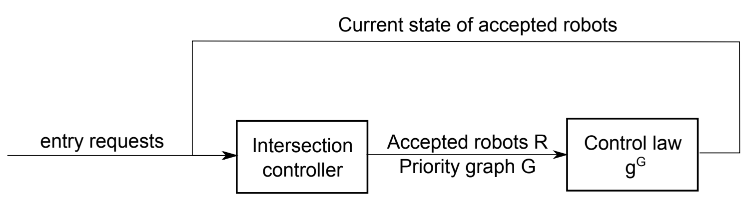

Figure 5: Scheme of the overall coordination system.

Figure 5 displays a scheme of the overall coordination system. Robots accepted in the control area are controlled by the control law g^G according to priorities G assigned by the intersection controller. Each robot is sequentially accepted into the control area by the intersection controller if it can go through the intersection at maximum throttle command, and if the travelled states are brake safe. As a result, if there is no control/sensing uncertainty and no unexpected event, the control law will always return $\overline{u}$. However, the purpose of the control law is precisely to handle uncertainty implicitly and to be able to decelerate at some point if necessary to avoid collisions or priority violation. This justifies feeding back into the intersection controller the current state of the robots, that can much deviate from the trajectories under perfect control law.

5. Illustration of the results through simulations

The following simulations provide qualitative results on the benefits of the proposed priority-based coordination system which combines a global planning approach (for priority assignment) and a feedback control law (for control under assigned priorities). The implementation of the simulator is completely centralized: communication aspects are not considered here.

5.1. The simulation setting

The algorithms presented in this paper have been implemented into a simulator coded in Java. Only straight paths are implemented and all robots are supposed to be circle-shaped with a common diameter D. This allows an easy computation of the obstacle region. Note that there is a finite set of possible paths for robots. As a result, the collision region between each couple of paths can be precomputed once and for all during the design phase of the intersection controller. Robots are generated at the origin of each path randomly at a constant rate. When generated, a robot i is positioned with zero velocity at the coordinate 0 of the path, or if there is already a robot j at position $x_j \leq D$, i is positioned at the coordinate $x_j - D$. As noted in [2], maximizing the velocity of robots in the intersection minimizes the time spent within the collision region, yielding a better performance. Hence, to ensure that robots have a maximum velocity within the collision region, the entry of the control area is defined to be far enough from the collision region (in the simulation videos we obviously see the robots that are not already accepted in the control area stop way before potential collision configurations). The intersection controller maintains a predicted trajectory for robots accepted in the control area. This predicted trajectory corresponds to the virtual trajectory defined in Equation (36). For each robot i, the predicted trajectory is initialized when it enters the control area. When the robot is accepted, its predicted trajectory assumes maximum throttle command. However, due to unpredicted events (e.g., a robot stops in the control area), the predicted trajectory may need to be updated. An update of the predicted trajectory of robots in the control area is carried out periodically (every 20 time slots in the presented simulations). If the current state of robots in the control area is s and the current priority graph is G, an update of the predicted trajectory consists of computing the trajectory $\tau \mapsto \Phi(\tau, s, g^G)$. This update process is key because the entry management is efficient only if the available predicted trajectories for robots already accepted in the control area are precise enough.

5.2. Simulations under deterministic control

The experimental intersection is depicted in Figure 6. It is composed of eight straights paths. The maximum velocity of robots is such that a robot at maximum velocity travels $D/2$ (one radius) during one slot. All robots share the same kinodynamic constraints with $\underline{u} = -\overline{u}$ and 20 slots are necessary to go from stop to full speed (and conversely). Hence, to ensure that robots are at maximum velocity when they reach the first potential collision configuration, the entry position is fixed at a distance $6D$ from the first potential collision configuration. Symmetrically, the exit position is fixed at a distance $6D$ after the last potential collision configuration.

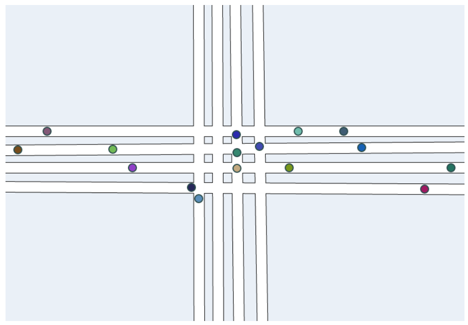

Figure 6: the intersection composed of eight straight paths used for simulations.

A video capture of the simulation for an arrival rate of 0.08 robots per time slot on each path is available[1]. One can observe that robots not accepted in the control area stop at a distance equivalent to 6 robots before the first potential collision configuration. In this simulation, there is no uncertainty, and the video capture confirms that in the absence of uncertainty, the presented algorithms result in robots always at maximum throttle command inside the control area. Finally, note that the entry management of the control area is not a first come first serve policy. Some robots requesting the entry before another robot may be accepted into the control area after that robot.

The latter phenomenon is more obvious in the video capture of the simulation for an arrival rate of 0.16 robots per time slot[2]. At such an arrival rate, queues are formed at the entry of the control area, but the size of the queues are not considered for processing the requests. Finally, note that queues are stable at this arrival rate which denotes an ergodic dynamics of the system.

At this point, note that the benefit of the priority-based approach is not visible, because in the absence of uncertainty, the control law under assigned priorities always returns $\overline{u}$. In the remaining simulations, we focus on particular uncertainties that may result in significant deviations from the predicted trajectory.

5.3. Simulations with unexpected events

Here, to illustrate the robustness of the proposed coordination system with respect to unexpected events, we consider a scenario in which robots may decide to brake within the control area unexpectedly. At the beginning of every time slot, each robot i may switch from a controlled regime under the control law g^G

[1] `http://www.youtube.com/watch?v=DXPICp4n2cU`
[2] `http://www.youtube.com/watch?v=fzgiXc6rgug`

to an unexpected deceleration under constant control $\underline{u}_i$, and vice versa, with probability transitions displayed in Figure 7. The probability values p, q are

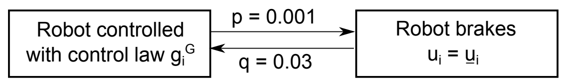

Figure 7: Non-deterministic transitions between control regimes

chosen arbitrarily, as the goal is is not to reproduce a realistic scenario but to test and validate the robustness of the approach. One may consider transitions to brake control regime as modelling some unexpected events subject to occur in applications to transportation systems such as a loss of communication abilities or a pedestrian crossing the road, both requiring the vehicle to slow down unexpectedly.

A video capture of the simulation for an arrival rate of 0.08 robots per time slot on each path is available here[3]. Even if some robots stop within the control area, other robots adapt and brake if necessary thanks to the control law. In contrast with simulations under deterministic control, the control law is useful here and enables to handle robots slowing down unexpectedly. No collision occurs during the simulation, the control law is effectively safe and robust with regards to brake application. We see that the priorities are satisfied, that no collision occurs, and that all vehicles eventually exit the intersection, although the trajectory may be very far from the trajectory under perfect control law.

6. Conclusions and perspectives

The key contribution of the present work is the introduction of a priority-preserving control law in Section 3. It both ensures collision avoidance and the respect of assigned priorities (see Theorem 1). Priorities need to be assigned: this is the topic of Section 4 where an overall coordination system is proposed. It is inspired from previous works on prioritized motion planning. Every robot is accepted into the control area sequentially in a way that maximizes its velocity through the intersection. When the robot is accepted, it is assigned the lowest priority among robots already accepted, which yields acyclic priorities, and thus ensures liveness as long as priorities are respected (see Theorem 3). The originality of the work is that the proposed coordination system combines a global motion planning approach for priority assignment, and a feedback motion planning approach for control under assigned priorities. The presented simulations illustrate the robustness of the algorithms with an example of unexpected events that make the robots drift far away from the trajectory that they would have followed in the absence of such events.

Finally, the presented work opens up new avenues. First of all, a possible distributed implementation of the proposed control law should be investigated.

[3] `http://www.youtube.com/watch?v=2-90q9aiEfg`

Indeed, for each robot, applying the control law only requires the knowledge of the state of neighboring robots. As a result, once priorities are assigned, robots could interact locally and there would be no need for large amounts of data being exchanged. This would be an interesting property from a practical viewpoint as in current wireless communication systems delays are unavoidable and the range is limited. The priority graph would then be seen as the minimal supervisory information ensuring that the individual actions of robots serve the overall goal.

Secondly, the priority assignment policy presented in Section 4 is rather naive. Indeed, queues of robots are not considered when processing the requests. Future work should focus on devising more complex priority assignment policies considering all robots at the proximity of the control area. This could be done adapting some of the many works already available on the optimization of prioritized motion planning methods (see e.g., [24, 25]). Moreover, in the context of a network of intersections in transportation networks, queueing network control systems already applied for traffic signal control (see, e.g., [26, 27, 28]) should be investigated.

Lastly, future works should focus on the design of a control law that is aware of sensing and control uncertainty. [18] has considered a two-vehicles coordination system and has evidenced that the concept of non-deterministic information state [29] (that consists of maintaining a set of possible states through time based on past history of observation and control) should prove useful for guaranteeing safety under sensing and control uncertainty. More precisely, a separation principle has been proved. We would like to extend this work to a multiple robot coordination system in the future.

[1] I. Zohdy, H. Rakha, Optimizing driverless vehicles at intersections, in: 10th ITS World Congress Vienna, Austria, 2012.

[2] K. Dresner, P. Stone, A multiagent approach to autonomous intersection management, Journal of Artificial Intelligence Research 31 (2008) 591–656.

[3] K. Kant, S. W. Zucker, Toward efficient trajectory planning: The path-velocity decomposition, The International Journal of Robotics Research 5 (3) (1986) 72–89.

[4] S. M. LaValle, Planning Algorithms, Cambridge University Press, Cambridge, U.K., 2006, available at http://planning.cs.uiuc.edu/.

[5] S. Leroy, J. P. Laumond, T. Simeon, Multiple path coordination for mobile robots: A geometric algorithm, in: In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI, 1999, pp. 1118–1123.

[6] T. Lozano-Perez, Spatial planning: A configuration space approach (1980).

[7] T. Fraichard, C. Laugier, Planning movements for several coordinated vehicles, in: Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89. Proceedings., IEEE/RSJ International Workshop on, 1989, pp. 466 –472. `doi:10.1109/IROS.1989.637945`.

[8] R. Ghrist, S. M. Lavalle, Nonpositive curvature and pareto optimal coordination of robots, SIAM Journal on Control and Optimization 45 (2006) 1697–1713. doi:10.1137/040609860.
URL http://dl.acm.org/citation.cfm?id=1237592.1237600

[9] M. Erdmann, T. Lozano-Perez, On multiple moving objects, Algorithmica 2 (1-4) (1987) 477–521.

[10] J. P. Van Den Berg, M. H. Overmars, Prioritized motion planning for multiple robots, in: Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on, IEEE, 2005, pp. 430–435.

[11] O. Mehani, 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, EUROCAST'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 1159–1166.
URL http://dl.acm.org/citation.cfm?id=1783034.1783191

[12] J. Peng, S. Akella, Coordinating multiple robots with kinodynamic constraints along specified paths, in: International Journal of Robotics Research, Springer-Verlag, 2002, pp. 221–237.

[13] S. G. Loizou, K. J. Kyriakopoulos, Closed loop navigation for multiple holonomic vehicles, in: Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, Vol. 3, IEEE, 2002, pp. 2861–2866.

[14] N. Vlassis, A concise introduction to multiagent systems and distributed artificial intelligence, Synthesis Lectures on Artificial Intelligence and Machine Learning 1 (1) (2007) 1–71.

[15] K. E. Bekris, K. I. Tsianos, L. E. Kavraki, Safe and distributed kinodynamic replanning for vehicular networks, Mobile Networks and Applications 14 (3) (2009) 292–308.

[16] J. Gregoire, S. Bonnabel, 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.

[17] D. Angeli, E. D. Sontag, Monotone control systems, Automatic Control, IEEE Transactions on 48 (10) (2003) 1684–1698.

[18] D. Del Vecchio, M. Malisoff, R. Verma, A separation principle for a class of hybrid automata on a partial order, in: American Control Conference, 2009. ACC '09., 2009, pp. 3638 –3643. doi:10.1109/ACC.2009.5160025.

[19] P. O'Donnell, T. Lozano-Periz, Deadlock-free and collision-free coordination of two robot manipulators, in: Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on, 1989, pp. 484 –489 vol.1. doi:10.1109/ROBOT.1989.100033.

[20] A. Colombo, D. Del Vecchio, Efficient algorithms for collision avoidance at intersections, Hybrid Systems: Computation and Control.

[21] H. Kowshik, D. Caveney, P. Kumar, Provable systemwide safety in intelligent intersections, Vehicular Technology, IEEE Transactions on 60 (3) (2011) 804 –818. doi:10.1109/TVT.2011.2107584.

[22] T. Fraichard, H. Asama, Inevitable collision states - a step towards safer robots?, Advanced Robotics -Utrecht- 18 (10) (2004) 1001–1024.
URL http://hal.inria.fr/inria-00182063

[23] E. C. Kerrigan, Robust constraint satisfaction: Invariant sets and predictive control, Ph.D. thesis, PhD thesis, Cambridge (2000).

[24] T.-C. Au, N. Shahidi, P. Stone, Enforcing liveness in autonomous traffic management, in: Proceedings of the Twenty-Fifth Conference on Artificial Intelligence, 2011, pp. 1317–1322.

[25] S. Akella, S. Hutchinson, Coordinating the motions of multiple robots with specified trajectories, in: Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on, Vol. 1, 2002, pp. 624 – 631 vol.1. doi:10.1109/ROBOT.2002.1013428.

[26] P. Varaiya, The max-pressure controller for arbitrary networks of signalized intersections, in: Advances in Dynamic Network Modeling in Complex Transportation Systems, Springer, 2013, pp. 27–66.

[27] T. Wongpiromsarn, T. Uthaicharoenpong, Y. Wang, E. Frazzoli, D. Wang, Distributed traffic signal control for maximum network throughput, in: Proceedings of the 15th international IEEE conference on intelligent transportation systems, 2012, pp. 588–595.

[28] J. Gregoire, E. Frazzoli, A. de La Fortelle, T. Wongpiromsarn, Capacity-aware back-pressure traffic signal control (2013).

[29] S. M. LaValle, S. A. Hutchinson, Evaluating motion strategies under nondeterministic or probabilistic uncertainties in sensing and control, in: Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, Vol. 4, IEEE, 1996, pp. 3034–3039.