

Priority-based coordination of mobile robots

Jean Gregoire

▶ **To cite this version:**

Jean Gregoire. Priority-based coordination of mobile robots. Other [cs.OH]. Ecole Nationale Supérieure des Mines de Paris, 2014. English. NNT : 2014ENMP0023 . tel-01081216

HAL Id: tel-01081216
https://pastel.hal.science/tel-01081216

Submitted on 7 Nov 2014

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.

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale n° 432 : Sciences des Métiers de l'Ingénieur

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure des Mines de Paris

Spécialité: "Informatique temps réel, robotique et automatique"

présentée et soutenue publiquement par

Jean Gregoire

le 29 septembre 2014

Coordination de robots mobiles par affectation de priorités

Priority-based coordination of mobile robots

Directeur de thèse: **Arnaud de La Fortelle**
Co-directeur de thèse: **Silvère Bonnabel**
Président du jury: **Denis Gillet**

Jury

Mme Domitilla Del Vecchio,	Professeur assistante, MIT	Rapporteur
M. Thierry Fraichard,	Chargé de recherche, INRIA Grenoble Rhône-Alpes	Rapporteur
M. Silvère Bonnabel,	Maître Assistant, MINES ParisTech	Examinateur
M. Emilio Frazzoli,	Professeur, MIT	Examinateur
M. Denis Gillet,	Maître d'enseignement et de recherche, EPFL	Examinateur
M. Arnaud de La Fortelle,	Professeur, MINES ParisTech	Examinateur
M. Jean-Paul Laumond,	Directeur de Recherche, LAAS	Examinateur

MINES ParisTech
Centre de Robotique (CAOR)
60 Bd Saint Michel, 75006 Paris, France

THÈSE

Abstract

Since the end of the 1980's, the development of self-driven autonomous vehicles is an intensive research area in most major industrial countries. Positive socio-economic potential impacts include a decrease of crashes, a reduction of travel times, energy efficiency improvements, and a reduced need of costly physical infrastructure. Some form of vehicle-to-vehicle and/or vehicle-to-infrastructure cooperation is required to ensure a safe and efficient global transportation system. This thesis deals with a particular form of cooperation by studying the problem of coordinating multiple mobile robots at an intersection area. Most of coordination systems proposed in previous work consist in planning a trajectory and to control the robots along the planned trajectory: that is the plan-as-program paradigm where planning is considered as a generative mechanism of action. The approach of the thesis is to plan priorities – the relative order of robots to go through the intersection – which is much weaker as many trajectories respect the same priorities. More precisely, priorities encode the homotopy classes of solutions to the coordination problem. Priority assignment is equivalent to the choice of some homotopy class to solve the coordination problem instead of a particular trajectory. Once priorities are assigned, robots are controlled through a control law preserving the assigned priorities, i.e., ensuring the described trajectory belongs to the chosen homotopy class. It results in a more robust coordination system – able to handle a large class of unexpected events in a reactive manner – particularly well adapted for an application to the coordination of autonomous vehicles at intersections where cars, public transport and pedestrians share the road.

Keywords: mobile robots, multi robot systems, motion planning, coordination space, priority graph, homotopy class, safety, robustness, hybrid architecture

Acknowledgment

Special thanks to my thesis directors Arnaud de La Fortelle – who hired me even though I criticized the city he lives in with his wife during the job interview – and Silvère Bonnabel – who supposedly accepted to work with me because we are both from Marseille. Discussions with you around drawings and equations have been dreamy.

Dreamy has also been the celebratory cocktail after my defense organized by the woman of my life, Mélodie, so many thanks for this, so many thanks as well for our intense and illuminating debates around perfomativity and for all your support and the fairy you have brought me all along the thesis.

Fairy has not necessarily been the exact thing that I have brought my office-mates the past three years. You – literally – gave me reasons to think however that you have appreciated my coordinated smarties at their fair value. Vincent, Raoul, Amaury, Eva, Edgar, Tao Jin, Philippe, Fernando, Jean-François, Fabien, Sébastien, thanks a lot for your happiness, your savoir vivre, and our – more or less – deep discussions.

All my thanks also for the two incredible ski trips that I had with the lab. Thanks in particular to Vincent and Sylvain, ski and after-ski will be remembered. Thanks also to Arnaud, director of the lab as well as of my thesis, for accepting to come with us despite our insane cruelty and his defeat during snow battles.

I complained a few times for not having the right to go to the canteen of the Ministry of Research for administrative reasons. It allowed me however – in addition to slightly enlarge by eating sandwiches instead of healthy meals – to spend a very nice lunch time in the garden of the school all along the thesis. Thanks in particular to Raoul, Axel, Martin, Xiangjun, Victorin, Tony, Zhuowei, Jorge, Laure, Olivier, Victor, Martyna, Bruno.

I must also thank Christine and Christophe, who learned me how to deal with administrative stuff like a superhero.

Having a coffee break has been so nice, always different, and let's say surprising,

thanks to the art of Thérèse in managing the cafeteria and connecting people.

So many thanks to my family for their support in my choice of doing research, and in particular to my mother [53] who inspired me doing research when I was just a kid, and to my father who constantly recalls me to work on what I love to do.

I also want to thank all the members of my thesis jury. It has been a pleasure that you have felt and shared my enthusiasm around my research work.

Many thanks also to newly arrived PhD students Xiangjun and Sébastien, for sharing this enthusiasm, discussing, continuing and going much ahead my research work.

So many thanks to my friends who took half a day off to listen to what I have loved to do the three past years and who transformed Mines ParisTech into the best cocktail bar in Paris ever just for the celebratory cocktail of my defense.

Finally, thanks to Mines ParisTech. After three years as an engineer student and three years as a PhD student, I need to cut the cord for some time. I believe you brought me as much knowledge, people, happiness, friendship and love as you could have.

To Mélodie and in honor of Fairy

Contents

Abstract i

Acknowledgment iii

1 Introduction 1
- 1.1 Industrial motivation . . . 1
- 1.2 Plan or react ? . . . 2
  - 1.2.1 Planning as a generative mechanism of action . . . 3
  - 1.2.2 Intelligence without planning . . . 6
  - 1.2.3 Plan as a resource to guide action . . . 8
- 1.3 Contributions . . . 12

I Priorities: definition and properties 15

2 Priorities: a geometric concept in the coordination space 21
- 2.1 The coordination space approach . . . 21
- 2.2 The priority relation . . . 26
  - 2.2.1 The completed obstacle region . . . 26
  - 2.2.2 The priority relation . . . 30
  - 2.2.3 The priority graph . . . 33

3 The coordination space under assigned priorities 37
- 3.1 Priorities: a homotopy invariant . . . 37
  - 3.1.1 Homotopy classes . . . 37
  - 3.1.2 Invariance of the priority graph . . . 40
- 3.2 Feasible priority graphs . . . 43
  - 3.2.1 Sufficient condition for priorities feasibility . . . 45
  - 3.2.2 Safety margin . . . 47
  - 3.2.3 Characterization of feasible priorities . . . 48
  - 3.2.4 Absence of deadlocks . . . 54

II Priority preserving control 57

4 Priority preserving control in the absence of inertia 63
- 4.1 A monotone control system . . . 63
- 4.2 The proposed control law . . . 64
  - 4.2.1 Priority preservation . . . 65
  - 4.2.2 Optimality . . . 68
  - 4.2.3 Liveness . . . 70

5 Priority preserving control under kinodynamic constraints **71**
5.1 A monotone control system 71
5.2 The proposed decentralized control law 73
5.2.1 Priority preservation 76
5.2.2 Robustness 78
5.2.3 Liveness 81

6 Robustness with respect to bounded noise **83**
6.1 Control model with bounded noise 84
6.2 Evolution of the non-deterministic information state 85
6.3 The proposed decentralized control law 87
6.3.1 Priority preservation 87
6.3.2 Robustness 88
6.3.3 Liveness 90

III Priority-based coordination **95**

7 Overall priority-based coordination system **101**
7.1 Three-layer architecture 101
7.1.1 The intersection controller 101
7.1.2 The behavior-based layer 103
7.1.3 The sequencer 104
7.2 Priority assignment 105
7.2.1 A simple priority assignment policy 105
7.2.2 Request processing liveness 107
7.2.3 Stability guarantees 107

8 Simulations **109**
8.1 Implementation aspects 109
8.2 Simulation results 111
8.2.1 Simulations under deterministic control 112
8.2.2 Robustness regarding unexpected deceleration 113
8.2.3 Robustness regarding bounded uncertainty 114
8.2.4 Stability guarantees under back-pressure control 115

Conclusions and perspectives **121**

Bibliography **127**

Appendix **143**

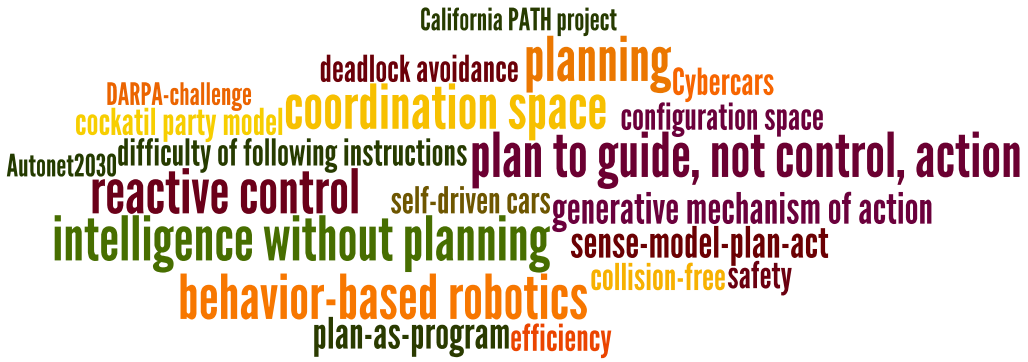
A Appendix of Chapter 2 **143**
A.1 Topology results 143
A.2 Proof of the South-West North-East completion 145
A.3 Proof of the existence of paths with maximal margin 146
A.4 Proof of the characterization of feasible priority graphs 148

B Priority preserving control: extension to feasible cyclic priorities **151**
B.1 In the absence of inertia 151
B.1.1 Existence 153
B.1.2 Safety 153
B.1.3 Optimality 153
B.1.4 Liveness 154
B.2 Under kinodynamic constraints 154
B.3 Under uncertainty in control and sensing 157

Résumés en Français **165**

Chapter 1

Introduction



Contents

1.1 Industrial motivation 1
1.2 Plan or react ? 2
1.2.1 Planning as a generative mechanism of action 3
1.2.2 Intelligence without planning 6
1.2.3 Plan as a resource to guide action 8
1.3 Contributions 12

1.1 Industrial motivation

The last decades have seen a number of projects addressing the automation of vehicles. The California PATH AHS project (1988-2003) was interested in making progress in automated highway systems [8, 59] with about 600 person-years of effort invested [121]. The European CityMobil project, finalized in 2011, addressed the integration of automated transport systems in the urban environment [2, 133] and the cooperation is continuing through CityMobil2 started in September 2012 for 4 years and involving 45 partners from system suppliers, city authorities, the research community and networking organizations [3]. The European interactIVe project, finalized in 2012, focused more on advanced driver assistance systems (ADAS) for collision avoidance by active intervention in intelligent vehicles [38]. The DARPA challenge, a prize competition for American autonomous vehicles, funded by the Defense Advanced Research Projects Agency, has also stimulated innovation and research in vehicles automation (see, e.g., [128]).

All these research projects are funded thanks to high expectations in terms of economic and social impacts. A recent report [137], jointly written by a consulting company together with an automotive research center, presents self-driving car as the next revolution in the automotive industry. Car automation is expected to decrease crashes, to reduce the need for physical costly infrastructures, to create new models of shared mobility, to reduce and anticipate travel times, to improve productivity, to improve energy efficiency: a combination of social and economic positive impacts. To the authors of the report, it is clear that such disruptive change in the automotive industry opens opportunities for new players and requires all companies of the sector to embrace innovation or to be left behind.

As autonomous vehicles are now starting to be deployed, cooperation among autonomous vehicles and also between autonomous and human-driven vehicles is necessary. This is the topic of the European project Autonet2030 [4], just started in 2014. Many use cases require cooperation: lane change negotiation, overtaking, cooperative routing, or cooperative speed control. This thesis focuses on the coordination of autonomous vehicles at intersections. Two main goals motivate the research in this topic. The first one is to avoid crashes due to collisions that occur particularly at intersections and because of human error (the leading factor in most of road accidents [130, 95]). The second one is to enhance road traffic efficiency, given that intersections represent bottlenecks in the traffic network resulting in congestion, one of the major problems in today's metropolitan transportation networks. As the results provided in this thesis can be applied to multiple domains including self-driving cars, we will use the more generic term robot instead of vehicle. We consider the problem of coordinating a collection of cooperative mobile robots at an intersection area, that is a region of space with a high concentration of potential collisions. According to the taxonomy proposed in [39], we propose to build a strongly coordinated multi robot control system aiming at ensuring safety and efficiency at intersection areas.

1.2 Plan or react ?

Since the 1980's, there is strong debate in the research community on the place of planning and reactive control in the design of autonomous robots [36]. For a while, the dominant view in the Artificial Intelligence community was that all the intelligence of an autonomous robot lies in its planning capabilities. On the other hand, Brooks, with the introduction of Subsumption architecture [21], gave birth to a departure from the traditional planning approach, repudiating plans, convinced that intelligent autonomous robots can be designed through simple interconnected primitive reactive behaviors. Research work involving researchers from Robotics, Artificial Intelligence and Sociology, attempts to conciliate both camps by providing a new view on what planning is. Planning is proposedly considered as the generation of resources to guide action [6], not as a generative mechanism of action. In the sequel, the three approaches introduced above are presented in more details and

a literature review of the coordination of multiple robots is provided through the prism of the long-standing debate around the relative place of planning and reactive control.

1.2.1 Planning as a generative mechanism of action

The Sense-Model-Plan-Act paradigm is the traditional approach of robot control in Artificial Intelligence with four components executed in a serial fashion [90]. The sensing system receives raw sensor input. The sensing data is turned into a world model by the modeling system. Provided a world model, the planner is in charge of computing a sequence of actions in order to achieve some goal: this step is time consuming and requires reasoning about the future. Finally, a low-level controller executes the plan. This traditional approach to planning, referred as plan-as-program [6], considers planning as "a generative mechanism of action" [123] as the planner dictates the actions to take in the future.

Reservation-based autonomous intersection management One of the most known autonomous intersection management system, proposed by Dresner and Stone [35, 33] (see a screen-shot of the simulator in Figure 1.1), espouses the plan-as-program paradigm. The system is based on requests sent by vehicles to a central

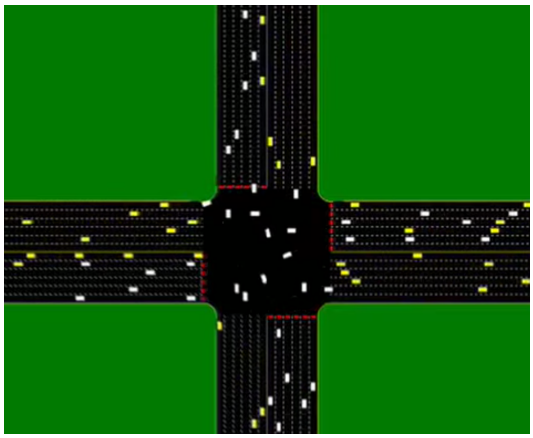


Figure 1.1: A screen-shot of the "Autonomous Intersection Management" simulator developed by the team of the Department of Computer Sciences at the University of Texas at Austin, USA.

agent in order to reserve space-time regions. Basically, a region of the space where lies the intersection is reserved for a given vehicle during a certain time interval. The central agent ensures that accepted reservations are compatible with each other. Safety is ensured as long as all vehicles respect the specification of their accepted requests, i.e., the specified starting time and velocity profile through the intersection. Hence, requests are the representation of the plan-as-program that robots

must execute. The approach has been widely studied with several incremental improvements (see, e.g., [34, 56, 11]). Variants of the approach have been developed by several authors. In particular, [26, 91] introduce critical points in order to improve the precision of the reservation system without increasing its complexity. In [142], environment variables are taken into account at the planning phase. Reference [64] proves that a reservation request can be processed in constant time provided the velocity profiles of vehicles is fixed. Experiments with real vehicles (Cybercars [1]) using a very simple reservation system are presented in [70].

Motion planning in the configuration space In Robotics, motion planning using the configuration space approach also espouses the plan-as-program paradigm. Reference [85] introduced the notion of configuration space in order to formalize the traditional motion planning problem. Basically, each dimension of the configuration space represents a degree of freedom, and there is an obstacle region in the configuration space which is the set of forbidden configurations for the robot, to model the presence of a static obstacle or some constraints due to the geometry of the robot (think of a robot with multiple arms). The traditional motion planning problem consists in finding a collision-free path in the configuration space from specified initial/goal configurations. A multi robot system can be considered as a generic robot whose configuration space is the Cartesian product of the configuration space of each robot [12, 115]. The obstacle region then contains forbidden configurations of each robot, plus forbidden composite configurations to account for possible inter-robot collisions. In this framework, the multi robot motion planning problem consists in finding a collision-free path from a composite start configuration (the start configuration of all robots) to a composite goal configuration (the goal configuration of all robots). Many methods have been devised in order to find a path in a constrained configuration space. For a system of two robots, a shortest path algorithm using the concept of visibility graph is proposed in [118]. In [124, 125], the authors show how to use dynamic programming to solve motion planning problems. Sampling based methods have also demonstrated their efficiency when the number of degrees of freedom is reasonable. Partial motion planning (see, e.g., [107, 15]) samples the action space and chooses the control to apply considering only a finite horizon, guaranteeing a bounded computation time. Other sampling methods include probabilistic roadmaps [67, 69], which have been applied to multiple robot motion planning in, e.g., [126]. An improvement of the probabilistic roadmaps, particularly useful for nonholonomic robots is the Rapidly-Exploring Random Trees [74] (an enhanced provably "optimal" version is proposed in [66]). All the above methods do not scale with the number of robots. Finding a path in the composite configuration space is of high computational complexity and becomes unfeasible in practice for a large number of robots [58].

In [65], a path-velocity decomposition allowing to reduce the problem's complexity is proposed. In this setting, each robot is assumed to move along a predefined path in its own configuration space and then the velocity profiles of the robots along

their assigned paths are optimized. The configuration of each robot boils down to its curvilinear position on its path and the configuration space of the whole system is 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. Such approaches based on the coordination space turn the coordination problem into the geometric problem of searching a collision-free path for a composite robot in a n-dimensional space where the obstacle region has a cylindrical shape [73, 81] (see Figure 1.2). Even though some authors highlight

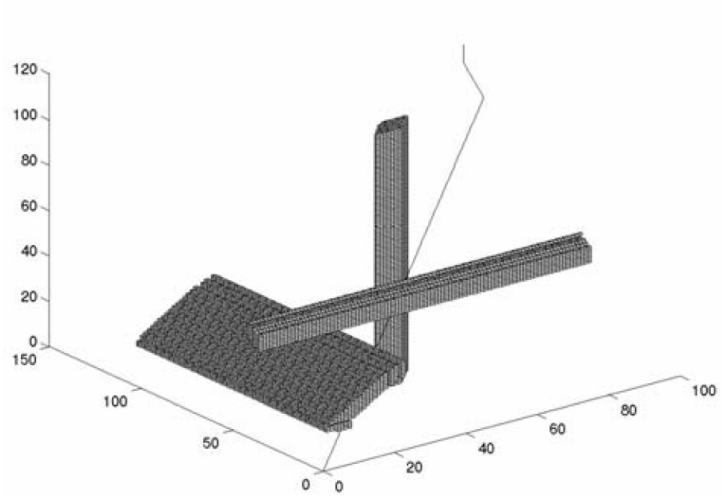


Figure 1.2: The cylindrical obstacle region in the coordination space for the coordination of three robots and a collision-free path (courtesy of [54]).

some cases where the approach fails [114], the approach has become standard in motion planning [72, 76, 81, 85, 40, 54]. For two-robot systems, optimal solutions have been proposed [120, 18, 22]. Reference [117] studies the time complexity of the coordination problem, defined as the completion time, i.e., the time for the last robot to reach its destination: lower and upper bounds are provided. The papers [47, 46] study the problem of finding Pareto-optimal trajectories, i.e., each robot tries to optimize its own particular objective function. They propose to first discretize the coordination space, and then to take advantage of the cylindrical structure to turn the coordination space into a negatively curved discrete space. Uniqueness of locally Pareto-optimal trajectories in each homotopy class of trajectories appears then as a mere consequence of the uniqueness of geodesics linking two points in a hyperbolic space. However, enumerating all locally optima in each homotopy class to find a globally optimal trajectory is a problem of high combinatorial complexity, and the authors point out the solution proposed is of interest only with a few robots and a low degree of intersection. The complexity of searching a trajectory in the configuration space led researchers to develop the so-called prioritized motion planning

method.

First introduced in [37], prioritized motion planning avoids the complexity of searching a trajectory in the n-dimensional coordination space. Instead of directly searching a trajectory for the composite robot in the coordination space, it consists in planning the trajectory of each robot sequentially. Each robot is mapped to a real number called the priority of the robot, and the trajectory of each robot is planned, in order of decreasing priorities: robots for which motion has already been planned are considered as dynamic obstacles [132]. The approach has been widely and successfully utilized. The assignment of the priorities is key to the quality of the planned trajectory, e.g., with regards to the delay due to coordination. For n robots, since priorities are sequential, there are $n!$ possible priority schedules. In [16, 17], a (randomized) search is proposed to optimize the prioritization scheme, [109] and [132] define simple heuristics for priority adjustment, and the heuristics of [23] dynamically updates the priorities of robots. Even if prioritized motion planning is not explicitly mentioned, the approach of [35, 91] for autonomous intersection management also belongs to the family of prioritized motion planning, because the trajectory of robots are planned sequentially. In [14], a prioritized planning is implemented using a multiagent system approach and taking into account communication aspects.

In [7, 103, 104], a collision-time formulation is proposed, it formulates the motion planning problem as a mixed linear programming (MILP) problem. Every robot is assumed to follow a path with a fixed velocity profile. Hence, the motion planning problem boils down to decide the starting time of each robot along the assigned trajectory (fixed path and velocity profile). The MILP formulation enables to solve the motion planning problem with efficient standard tools for MILP problems, so that as many as 20 robots can be coordinated, according to the authors.

On the difficulty of following instructions As noticed in [123], the main weakness of the plan-as-program approach resides in the inherent "difficulty of following instructions" in the face of environmental uncertainty and unpredictability. It has rapidly become admitted in the Artificial Intelligence community that unpredictability makes open-loop plan execution inefficient and leads to undesired behaviors. Replanning through time to account for new information is an attempt to treat this issue (see, e.g., [13, 14] for dynamic replanning of a multi robot system). However, planning is a time consuming task and constantly replanning makes difficult to respect real time execution constraints. These difficulties are at the origin of a completely opposite approach. Instead of considering that intelligence lies in the planning phase, a community of researchers initiated by the seminal work of Brooks [21] tried to design intelligent robots that do not rely on planning at all.

1.2.2 Intelligence without planning

Reactive and behavior-based robotics In [21], Brooks proposed the foundation of what became behavior-based robotics. The main source of novelty is to

abandon a centralized and centrally manipulated representation of the world [89]. Instead, the robot control system is layered with several behaviors, each one achieving and/or maintaining a specified goal, e.g., "avoid-obstacles", "go-home". A behavior has either absolutely no world model and no internal state in which case the behavior is purely reactive mapping sensor data to actions, or it has its own minimal internal state maintained only in order to achieve its own goal. Behaviors are simple enough to run in real-time, they run in parallel and are layered so that the capability of the system increases as new behaviors are introduced. The intelligence of the autonomous system is not necessarily obvious when looking at every individual behavior. However, so-called emergent behaviors – intelligence "in the eye of the beholder" [138] – originate from the large amount of interactions between behaviors in the environment. The belief of the research community advocating for behavior-based design is that their approach can scale to higher level of complexity than many researchers of the plan-as-program school assume and result in more efficient and robust systems than through traditional planning [138].

The "cocktail party" model is a simple example of such a reactive approach to the coordination of multiple robots. In this setting proposed in [86], a robot can only sense the surrounding objects, it knows its current and its target position, it can distinguish between static obstacle and robots and can sense the instantaneous motion of other robots. A reactive coordination of robots is proposed with only these capabilities and without mutual communication. The authors claim the obtained system demonstrates good performance and a remarkable robustness. The idea of the proposed algorithm is based on maze-searching techniques. Robots follow the boundary of static obstacles. For moving obstacles (including other robots), a collision front is build considering the maximal motion of the moving obstacles. Then, as for static obstacles, the boundary of the collision front is followed, ensuring collision avoidance. The term "cocktail party" is justified by the analogy with the behavior of a guest willing to talk to someone in a crowded place, such as a cocktail party. The guest travels between tables, chairs and other guests, planning his/her motion "on the fly". In [86], only the translation of robots in the plane is considered and robots are assumed to be able to stop instantly. These assumptions are relaxed in [100] where nonholonomic constraints are considered.

On the difficulty of deadlock avoidance under reactive schemes As noticed in [86, 100], deadlock avoidance is difficult to ensure in such a reactive control scheme. Reference [86] provides an example itself drawn from a previous work [115] and depicted in Figure 1.3. As noticed by Lumelsky, the task seems to be "impossible" unless some planning is carried out to coordinate the motion of robots (see comments in Figure 1.3). Reference [24] proposes a general characterization of system deadlocks as a situation where the following conditions hold:

- tasks claim exclusive control of the resources they require ("mutual exclusion" condition);

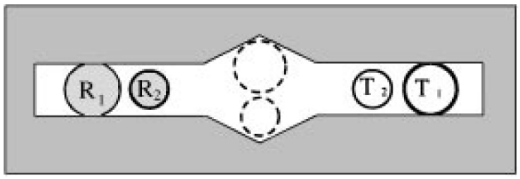


Figure 1.3: An example where planning is necessary to achieve a task. "In order for the circular robots R_1 and R_2 to reach their respective targets, their relative position need to be switched. The only way to do so is to move R_2 into one of the 'wedges' and then move R_1 through the other wedge. The task is clearly impossible unless the motion of both robots is closely coordinated in a centralized manner." (courtesy of [115])

- tasks hold resources already allocated to them while waiting for additional resources ("wait for" condition);
- resources cannot be forcibly removed from the tasks holding them until the resources are used to completion ("no preemption" condition);
- a circular chain of tasks exists, such that each task holds one or more resources that are being requested by the next task in the chain ("circular wait" condition).

To illustrate the notion, the authors give an example of traffic deadlock (see Figure 1.4). As noticed by the authors, in this example, resources are the space occupied by cars. The "mutual exclusion" condition holds as two cars cannot occupy the same region without colliding. The "wait for" condition also holds as cars need to move forward (to get the next the space region) before releasing the current space region. The "no preemption" condition holds as cars cannot disappear from the real space, and finally the "circular wait" condition is clearly visible in Figure 1.4. Later work refined the notion of deadlock in resource allocation systems and studied the complexity of deadlock avoidance/resolution (see, e.g., [10, 78] and [110, 111, 113] for a specific study focused on multiple robot systems).

The deadlock avoidance problem mirrors the inability (by design) of reactive systems to carry out look-ahead to make better choices to accomplish actions. In the late 1980's, a research movement initiated by Agre and Chapman attempted to reconcile the camp of plan-as-program with the school of reactive control by asking the question: "what are plans for ?" [6].

1.2.3 Plan as a resource to guide action

Plans should be used to guide, not control, action First of all, inspired by previous work of social scientists (see references therein [6]), Agre and Chapman proposed to retire the term "plan execution" advocated by the plan-as-program camp, considered as "prejudicial", and to use a more neutral term: "using a plan".

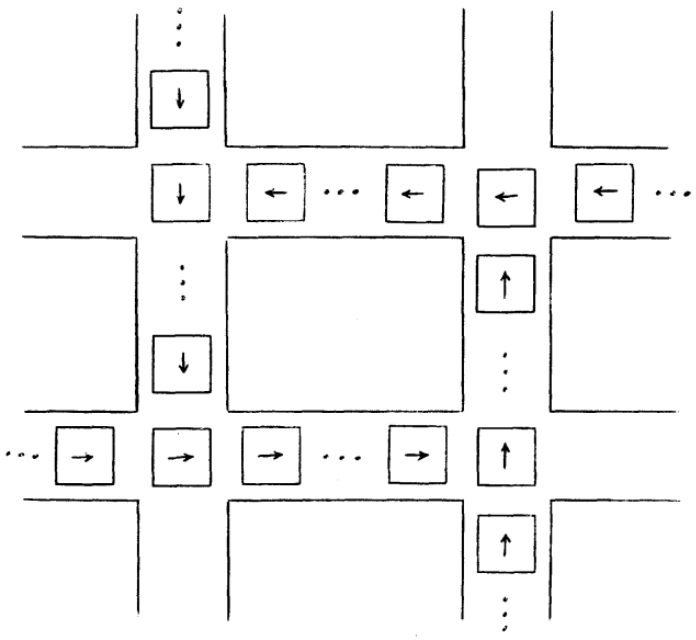


Figure 1.4: An example of traffic deadlock with a circular chain of vehicles blocking each other (courtesy of [24])

They noticed that this terminology consideration raises new questions. First, "what can one do with a plan besides executing it ?". Second, if plan users are able to use plans sensibly rather than simply executing them, what are the implications on the representation of plans (how a plan looks like) and on the generation of plans (how to devise a plan easy to use in a sensible way by plan users ?). Conceptually, they propose to consider planning as the devising of resources to guide action. Plans are not executed but they are interpreted. Plans are a resource among others to decide the action to execute. Planning tasks are executed in parallel and asynchronously in order to retain a reactive quality. Plans are here to let the system be more goal-directed, to enhance performance, not to dictate action.

Gradient fields as a guide to action As first noticed in [102], gradient fields are an example of the new kind of plan – a resource to guide action – proposed in [6]. As depicted in Figure 1.5, there is no explicit "traditional" plan. This plan can, instead, be interpreted as: "follow the arrows to reach the goal". Such plan is much more flexible than a traditional plan-as-program: at any point of time, a robot may decide not to follow the arrows for some reason (the sensors of the robot suddenly detect an obstacle that seems have not been detected at the moment of the gradient field computation). Nevertheless, the robot can still use the gradient field in the future no matter where it is currently located. In many scenarios, the robot will reach the goal without replanning, i.e., without a new time consuming computation

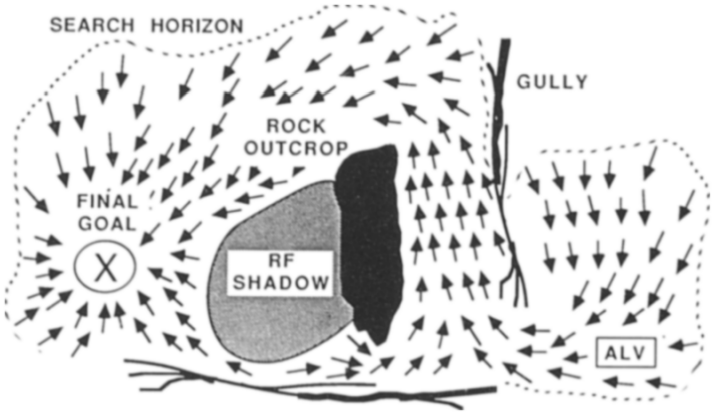


Figure 1.5: A gradient field to guide a mobile robot. The mission of the robot is to get from its current location to the "final goal" location, bypassing a gully and avoiding a large rock. Moreover, along the maneuver, the robot needs to maintain communication with a remote radio tower. To this purpose, the "shadow" of the rock should also be avoided (courtesy of [102]).

of the vector field. Vector fields approaches for single robot (see, e.g., [112]) have been adapted for the coordination of holonomic robots in [83]. In [31] a totally distributed version is proposed, the approach of [32] also applies to nonholonomic robots, and in [105] sensing and communication constraints are taken into account.

Traffic signals as a guide to action Traffic signals are an effective way to coordinate competing traffic flows at intersections. To this purpose, they alternate the right of way of users (cars, buses, pedestrians). A particular set of feasible simultaneous rights of way, called a phase, is decided for a certain period of time [101] (see Figure 1.6). In an intersection ruled by a traffic signal, the traffic signal does not dictate actions to users. A pedestrian is not obliged to cross the road even if he/she is given the right of way. Vehicles are invited to cross the intersection when they have the right of way. However, they do not have to follow a precise assigned velocity profile, and if some unexpected event occurs like a pedestrian crossing the road without the right of way, the vehicle should stop as it is much more important to avoid pedestrians than to cross the intersection rapidly. Hence, traffic signals are a good example of a planning process that consists in providing resources to guide actions. Without such a resource, it would be difficult for vehicles to coordinate. Rules can also be decided in advance or be displayed using traffic signs, and again, they constitute the resources to guide action just like the traffic signal, and the design of rules can be considered as a planning process.

A valuable property of traffic signals compared to static rules is that the traffic signal can be controlled in order to enhance efficiency. Controlling a traffic light consists in designing rules to decide which phase to apply over time. It is interesting to

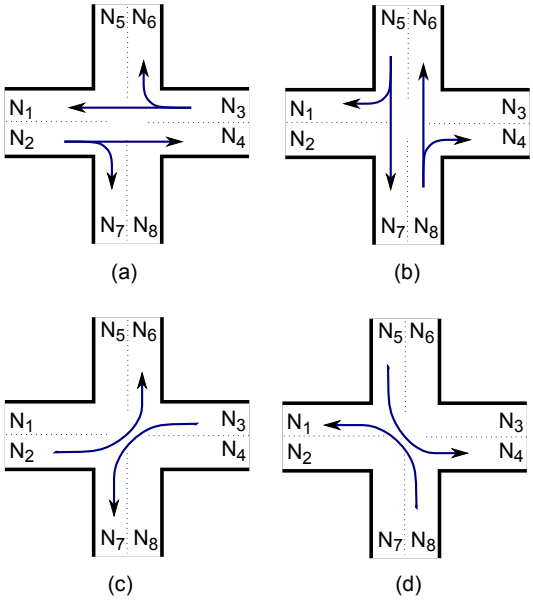


Figure 1.6: A typical set of feasible phases at a junction.

see that just like any planning process, traffic signal optimization is a complex time consuming task. Pre-timed policies activate phases according to a time-periodic pre-defined schedule. There is much previous work on designing optimal pre-timed policies, e.g., [93]. However, such policies are not efficient under changing arrival rates which require adaptive control. Most used adaptive traffic signal control systems include SCOOT [62], SCATS [84], PRODYN [57], RHODES [94], OPAC [42] or TUC [30]. These systems update some control variables of a configurable pre-timed policy on middle term, based on traffic measures, and apply it on short term. Control variables may include phases, splits, cycle times and offsets [101]. Such algorithms may differ in the way optimization is carried out (e.g., mixed-integer linear programming [43], dynamic programming, exhaustive enumeration) and in the modeling approach (e.g., queuing network model [98, 99], cell transmission model [82], store-and-forward [5], petri nets [29]). Many major cities currently employ these systems which proved to be able to yield various benefits, including travel time and fuel consumption reduction, as well as safety improvements [119]. More recently, based on the seminal paper [127], feedback controls have been proposed both in the case of deterministic arrivals [135], or stochastic arrivals [134, 140, 79]. Time is slotted and at every time slot, a feedback controller decides the phase to apply based on current queue lengths estimation. This requires real-time queue length measures, but it enables to be much more reactive than other traffic controllers and to have stability guarantees. Reference [127] introduced the so-called back-pressure control which computes the control to apply based on queue lengths, and can achieve provably maximum stability. This algorithm was originally applied to wireless communication networks [96], and some effort has been required to apply

the approach in the context of a network of intersections [134, 140, 52, 51]. A key feature of this algorithm is that it can be completely distributed over intersections, in the sense that it can be implemented by running an algorithm of complexity $\mathcal{O}(1)$, requiring only local information, at each intersection.

1.3 Contributions

In this thesis, we advocate for the use of planning as the devising of resources to guide action. Such approach is more than designing a system with both planning and reactive abilities. It requires rethinking what a plan for our system is. In [33], a plan is a set of granted reservation requests (composed of the starting time and the velocity profile), that vehicles must execute. In [132], the plan is simply the trajectory for each robot in the configuration space. A recent work [71] attempted to devise a coordination system for vehicles at intersections allowing some "freedom of action to cars" yet ensuring safety. Hence, the motivation is clearly to have a coordination with both planning and reactive abilities. However, the plan has quite the same representation as in traditional reservation-based systems. We believe that thinking plans as a resource to guide action should lead to a new representation of plans that should differ from traditional systems espousing the plan-as-program paradigm. What kind of plan to guide action can be designed for a multi robot system ?

The main contribution of this thesis is to propose a novel tool in multi robot motion planning: the priority graph. Roughly speaking, priorities describe a high-level coordination strategy: the relative order of robots. More precisely, they uniquely encode the homotopy classes of solutions to the multi robot coordination problem. This powerful theoretical tool is actionable to design a low complexity and robust priority-based coordination system. The planning process consists in assigning priorities. Under assigned priorities, robots can safely travel through the intersection in a reactive way provided all robots respect the assigned priorities. As planning priorities does not dictate a precise trajectory for robots through the intersection, but only provides useful concise resources for safe and efficient coordination, the system demonstrates valuable robustness properties in the face of environmental uncertainty and unpredictability. The thesis is organized as follows.

The first part presents the geometrical foundation of the priority-based approach using the standard coordination space framework. Priorities are formally defined and assigning priorities is provably equivalent to constrain the trajectory of robots to remain in a homotopy class of collision-free trajectories continuously deformable into each other. Assigning priorities does not plan a particular trajectory that robots must execute, yet it plans a higher-level coordination strategy describing the relative order of robots through the intersection: the priority graph. The priority graph can be considered as a unique meaningful representative of a homotopy class of trajectories. Planning priorities is a task of high combinatorial complexity as the set of possible priorities grows exponentially with the number of robots. The most

important feature that the priority assignment policy must demonstrate is that the assigned priorities are feasible, i.e., that robots respecting the assigned priorities will eventually go through the intersection. Roughly speaking, the assigned priorities should encode a "non empty homotopy class" of solutions to the coordination problem. Interestingly, given assigned priorities, there are two exclusive options: either the assigned priorities will inevitably lead robots to a deadlock configuration where a circular chain of robots block each other; or the assigned priorities are feasible and robots will provably never reach a deadlock configuration provided priorities are respected. As a consequence, deadlock avoidance can be completely solved at the priority assignment level. It is a valuable property motivating the use of priorities as a coordination resource to guide robots through the intersection as deadlock avoidance is difficult to guarantee in a reactive manner.

In contrast with the first part which has a quite mathematical – more precisely, geometrical – flavor with little care about control issues, the second part shows how to use priorities to guide robots through the intersection with control laws configured by the priority graph and ensuring priority preservation. Most importantly, under assigned priorities, for each pair of robots, there is not two but only one strategy to avoid collisions: the robot with lower priority must decelerate in favor of the robot with higher priority. As a consequence, the combinatorial complexity of multi robot control [25] is avoided, and priority preserving control is of polynomial complexity, thus allowing real-time implementation. Moreover, the proposed control law demonstrates a quite novel robustness property in the presence of inertia. Robots may indeed safely brake at any point of time without violating priorities, in particular without colliding, which is referred as brake safety. This can be useful to handle unexpected events requiring braking like a pedestrian crossing the road. It is a quite novel property with regards to previous work as the standard plan-as-program approach constrains robots to track precisely a planned reference trajectory and thus does not allow a robot to brake if necessary to handle some unexpected event. Finally, the control scheme proposed in Chapter 5 and 6 is decentralized. Each robot can compute the output of the control law independently without agreement with other robots through communication links. This benefit results from the prior agreement on the priority graph carried out at the planning level and requiring of course some form of communication.

The final part of the thesis proposes a priority-based coordination system adopting a three-layer control architecture. It has a more engineering flavor, specifying how priorities can be assigned dynamically as new robots arrive at the intersection and how to integrate priority preserving control proposed in the second part. A central agent, the intersection controller, constitutes the deliberative layer and assigns priorities. Robots implement several behaviors executed in parallel each one achieving/maintaining a specified goal, they constitute the reactive behavior-based layer. Behaviors include path following, moving forward, not entering the intersection before being accepted by the intersection controller, avoiding pedestrians, and of course respecting priorities which implements the priority preserving control law proposed in the second part of the thesis. Robots communicate asynchronously

with the intersection controller through the sequencing layer to request the right of way and be assigned assigned priorities. The sequencing layer interfaces with the behavior-based layer by activating/deactivating/configuring behaviors. The behavior-based layer takes full benefit of the brake safety property. Some behavior, e.g., the behavior ensuring pedestrian avoidance, may indeed require a robot to brake at any point of time with the guarantee that it will not result in a priority violation. Therefore, the coordination system demonstrates significant robustness as it can handle a large class of unexpected events – all events requiring braking – without changing priorities, i.e., without replanning. Priority-based coordination combines the efficiency of traditional planning approaches as complex scheduling can be encoded by the priority graph – much more complex scheduling than using traffic signals – as well as the ability to handle a large class of unexpected events in a reactive manner.

Part I

Priorities: definition and properties

Introduction

The present part constitutes the geometrical foundation of the priority-based approach proposed in this thesis. Priorities at road intersections are a well known concept aiming at organizing traffic. Signs, signals, markings are used to inform the users about who has the right to go first, or equivalently who has "priority" [139]. Convinced that the coordination space approach [97] is a convenient mathematical formulation of the coordination problem, we propose here a formal definition of priorities as a new concept in the coordination space. In the coordination space approach, a multi robot system composed of n robots traveling along fixed paths is considered as a composite robot evolving in a n-dimensional space called the coordination space [97, 73]. Potential inter-robot collisions requires the composite robot to avoid an obstacle region in the coordination space. The obstacle region has a cylindrical structure (see Figure 1.2). In traditional motion planning, the coordination problem is reduced to finding a feasible path in the coordination space (see the collision-free path in Figure 1.2). It looks like the notion of priorities is completely lost. In this part, we provide theoretical tools in the coordination space in order to endow the coordination space approach with a concept of priority. The idea is that a collision-free path in the coordination space necessarily lies on one side or on the other side with respect to each collision cylinder. Deciding on which side to pass with respect to each collision cylinder is equivalent to deciding the relative order of robots to go through the intersection and constitutes the discrete part of the coordination problem that we refer as priority assignment. Respecting assigned priorities does not require robots to follow a precise path in the coordination space as many collision-free paths respect the same priorities, or equivalently, lie on the same side with respect to collision cylinders. Hence, it is possible to assign priorities, yet retaining some individual freedom of action to robots. More precisely, the result of this part enable to go one step ahead in the understanding of the structure of the solutions to the coordination problem. Previous work noticed the existence of homotopy classes of feasible paths in the coordination space [47], and this part demonstrates that priorities are a unique meaningful representative of homotopy classes – they uniquely encode homotopy classes.

Sketch of the part Chapter 2 starts by exposing the coordination space approach, introducing assumptions and notations. Priorities are defined as a binary relation between robots induced by a collision-free path in the coordination space. As the coordination space is thus endowed with a priority concept, Chapter 3 studies the structure of the coordination space under assigned priorities. It is proved that all paths respecting the same priorities are continuously deformable into each other, forming a homotopy class. Finally, the deadlock avoidance problem is shown to be solved by assigning so-called feasible priorities which are characterized. This part motivates the use of priorities as a plan to guide robots through the intersection.


deadlock
homotopy class
feasible priorities
relative order
acyclic priorities
topology
coordination resource to guide robots
priority
erosion
south-west completion
high-level coordination strategy
completed obstacle region
dilatation
invariance
assign priorities
deadlock avoidance
homotopic paths
safety margin

Table of Contents

2 Priorities: a geometric concept in the coordination space 21

2.1 The coordination space approach 21

2.2 The priority relation 26

2.2.1 The completed obstacle region 26

2.2.2 The priority relation 30

2.2.3 The priority graph 33

3 The coordination space under assigned priorities 37

3.1 Priorities: a homotopy invariant 37

3.1.1 Homotopy classes 37

3.1.2 Invariance of the priority graph 40

3.2 Feasible priority graphs 43

3.2.1 Sufficient condition for priorities feasibility 45

3.2.2 Safety margin 47

3.2.3 Characterization of feasible priorities 48

3.2.4 Absence of deadlocks 54

Chapter 2

Priorities: a geometric concept in the coordination space

Contents

2.1 The coordination space approach 21
2.2 The priority relation 26
2.2.1 The completed obstacle region 26
2.2.2 The priority relation 30
2.2.3 The priority graph 33

Sketch of the chapter Section 2.1 is quite expository, it recalls the basics of the coordination space approach providing the main assumptions and notations. Section 2.2 endows the coordination space approach with a priority concept, defining the priority relation as well as the priority graph induced by a feasible path in the coordination space.

2.1 The coordination space approach

Consider the problem of coordinating the motion of a collection of robots $\mathcal{R}$ in a two-dimensional space. Every robot $i \in \mathcal{R}$ 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 2.1). $x := (x_i)_{i\in\mathcal{R}}$ 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, first introduced in [97] and which has become a standard tool [73]. This approach is often referred as path-velocity decomposition. It reduces the problem's complexity as each robot has now only one degree of freedom. For an application to autonomous vehicles at road intersections, this additional constraint seems particularly well adapted as the road network is strongly spatially organized (roads and lanes with markings). In the rest of the manuscript, $\{\mathbf{e}_i\}_{1\leq i\leq n}$ denotes the canonical basis of χ. Given a subset A of the topological space χ, ∂A refers to the boundary of A. We define the Minkowski sum as follows:

$$\forall x^0 \in \chi, \forall A \subset \chi,\ x^0 + A = \left\{x^0 + x : x \in A\right\} \tag{2.1}$$

$$\forall A, B \subset \chi,\ A + B = \{x + y : x \in A, y \in B\} \tag{2.2}$$

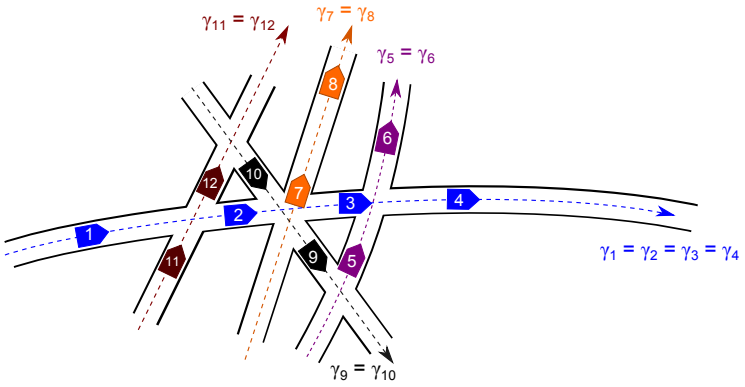


Figure 2.1: The fixed paths assumption. Every robot travels along an assigned path.

We will use the topology of infinity norm on $\chi \equiv \mathbb{R}^n$, so the parallelepiped $x^0 + (-r, r)^n$ is the open ball of radius $r > 0$ centered in $x^0 \in \chi$.

Some configurations must be excluded to avoid collisions between robots (see Figure 2.2). The obstacle region $\chi^{\text{obs}} \subset \chi$ is the open set of all collision configurations. Let $\kappa_{ij} \subset \mathbb{R}^2$ denote the set of configurations x where i and j collide. Let $\chi_{ij}^{\text{obs}} \subset \chi$ denote the set of (global) configurations x where i and j collide, we have:

$$\chi_{ij}^{\text{obs}} := \{x \in \chi : (x_i, x_j) \in \kappa_{ij}\} \tag{2.3}$$

We obviously take $\chi_{ii}^{\text{obs}} := \emptyset$.

Definition 2.1.1 (Obstacle region, Obstacle-free region)**.** *The obstacle region is the set $\chi^{\text{obs}} \subset \chi$ of configurations where a collision occurs for some $i, j \in \mathcal{R}$, i.e.,*

$$\chi^{\text{obs}} := \cup_{\{i,j\}} \chi_{ij}^{\text{obs}} \tag{2.4}$$

$\chi^{\text{free}} := \chi \setminus \chi^{\text{obs}}$ denotes the obstacle-free space.

By construction, χ_{ij}^{obs} is a cylinder (based on the plane generated by $\mathbf{e}_i$ and $\mathbf{e}_j$), and the obstacle region merely appears as the union of $n(n-1)/2$ cylinders [73] corresponding to as many collision pairs. Every cylinder χ_{ij}^{obs} is assumed to have an open bounded convex cross-section, i.e., κ_{ij} is open and bounded. The boundedness condition on χ^{obs} is rather technical but ensures the whole intersection lies in a bounded region. In particular, it implies that there exists a lower bound $\underline{x}^{\text{obs}} \in \chi^{\text{free}}$ and an upper bound $\overline{x}^{\text{obs}} \in \chi^{\text{free}}$ satisfying:

$$\forall i, j \in \mathcal{R}, \forall x \in \chi_{ij}^{\text{obs}},\ \underline{x}_i^{\text{obs}} < x_i < \overline{x}_i^{\text{obs}} \text{ and } \underline{x}_j^{\text{obs}} < x_j < \overline{x}_j^{\text{obs}} \tag{2.5}$$

A continuous application $\varphi : [0, 1] \to \chi$ will be called a path and we let $\text{Im}(\varphi)$ denote the set of values taken by φ:

$$\text{Im}(\varphi) := \{\varphi(t) : t \in [0, 1]\} \tag{2.6}$$

A partial order $\leq$ for configurations is defined as the product order of $\mathbb{R}^n$:

$$\forall x, y \in \chi, x \leq y \text{ if } \forall i \in \mathcal{R}, x_i \leq y_i \tag{2.7}$$

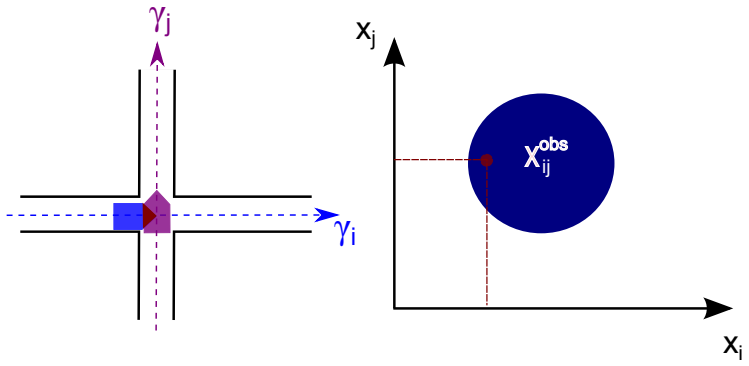


Figure 2.2: The left drawing depicts two paths with two robots in collision in the current configuration. The right drawing shows the obstacle region χ_{ij}^{obs} associated to the two paths (more precisely its cross-section along the plane generated by $\mathbf{e}_i$ and $\mathbf{e}_j$) and the collision configuration $x \in \chi_{ij}^{\text{obs}}$ corresponding to the collision of the left drawing.

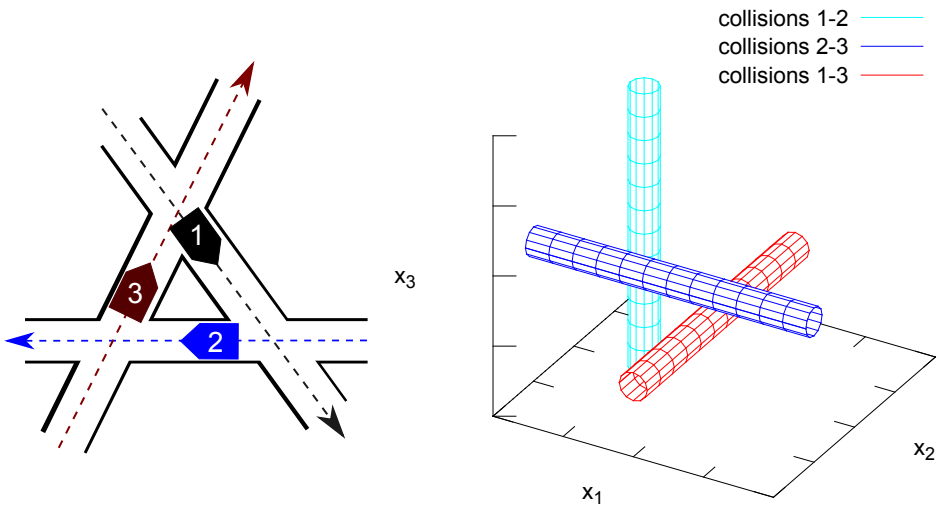


Figure 2.3: The right drawing shows the cylindrical structure of the obstacle region for the three-robot system of the left drawing. Each cylinder accounts for the possible collisions between each couple of robots. The right drawing of Figure 2.2 depicts the base of such cylinders.

Definition 2.1.2 (Feasible path)**.** *A feasible path is a non-decreasing collision-free path* $\varphi : [0,1] \to \chi^{\text{free}}$ *requiring no coordination beyond its endpoints, i.e., a path satisfying the following conditions:*

(a) φ *is non-decreasing:*

$$\forall t^1, t^2 \in [0,1],\ t^1 \leq t^2 \Rightarrow \varphi(t^1) \leq \varphi(t^2) \tag{2.8}$$

(b) φ *is collision-free:*

$$\text{Im}\,(\varphi) \subset \chi^{\text{free}} \tag{2.9}$$

(c) No coordination is required beyond its start point:

$$(\varphi(0) - \mathbb{R}^n_+) \subset \chi^{\text{free}} \tag{2.10}$$

(d) No coordination is required beyond its endpoint:

$$(\varphi(1) + \mathbb{R}^n_+) \subset \chi^{\text{free}} \tag{2.11}$$

We let $\Phi(\chi^{\text{free}})$ denote the set of feasible paths. Note that the two last conditions hold in particular for $\varphi(0) \equiv \underline{x}^{\text{obs}}$ and $\varphi(1) \equiv \overline{x}^{\text{obs}}$. The conditions of the above definition are more flexible and do not fix the endpoints. More importantly, we will only consider as feasible motions where robots never move backwards in the intersection area. It is a standard assumption as neither efficiency nor safety can be expected from robots moving backwards at an intersection area.

More generally, given a subset $C \subset \chi$, we let $\Phi(C)$ denote the set of non-decreasing paths satisfying $\text{Im}\,(\varphi) \subset C$, $(\varphi(0) - \mathbb{R}^n_+) \subset C$ and $(\varphi(1) + \mathbb{R}^n_+) \subset C$. This notation is coherent with the definition of $\Phi(\chi^{\text{free}})$ as the set of feasible paths. Using this notation, $\Phi(\chi)$ merely refers to the set of non-decreasing paths as the additional conditions obviously hold for $C \equiv \chi$. In the following, we provide three examples where the obstacle region can be computed analytically.

Example 2.1.1 (Two straight paths with circle-shaped robots)**.** Consider two circle-shaped robots of diameter D moving along straight and perpendicular paths. Assume that the curvilinear coordinate of each robot is 0 when the center of the robot is exactly at the paths intersection point. Then, the distance between the centers of robots is $d = x_i^2 + x_j^2$. As the diameter of robots is D, the configuration (x_i, x_j) is collision-free if and only if $d \geq D$, i.e., $x_i^2 + x_j^2 \geq D$. As a result, the obstacle region is $\chi^{\text{obs}} = \left\{x \in \chi : x_i^2 + x_j^2 < D\right\}$ as depicted in Figure 2.4.

Example 2.1.2 (Two perpendicular paths with rectangular robots)**.** Consider two rectangular robots i, j of lengths L_i, L_j and widths l_i, l_j along straight perpendicular paths. In the real space, there is a rectangular region of area $l_i \times l_j$ that can be occupied by only one robot, exclusively (see the red box in the left drawing of Figure 2.5). When a robot is at the the entry of this region (robot i in the left drawing of Figure 2.5), it needs to travel the length of the region plus its own

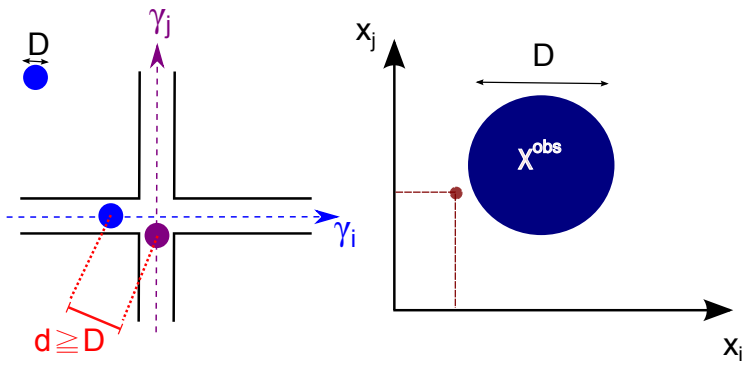


Figure 2.4: The obstacle region for two circle-shaped robots along straight perpendicular paths.

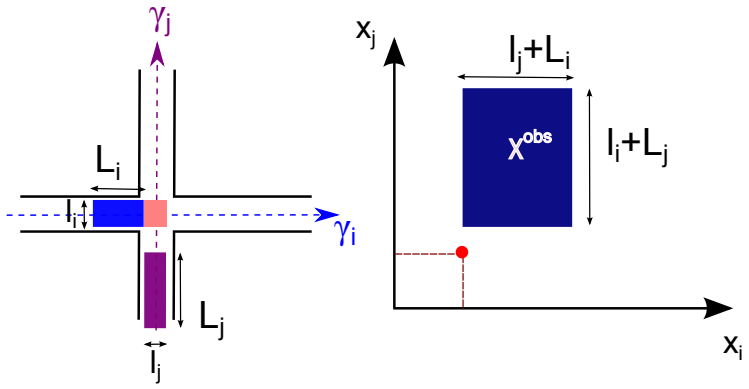


Figure 2.5: The obstacle region for two rectangular robots along straight perpendicular paths.

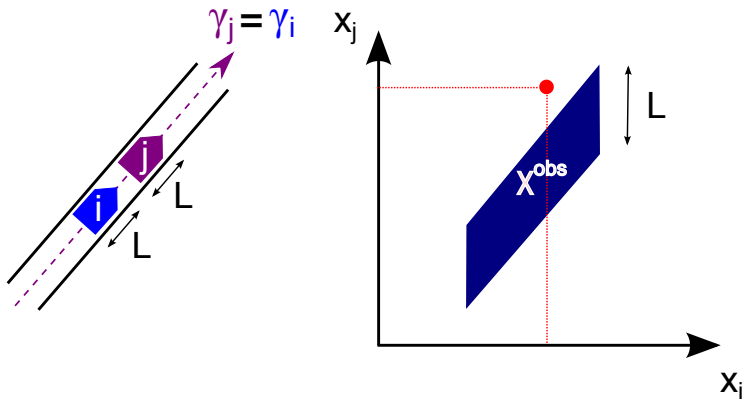


Figure 2.6: The obstacle region for two robots that follow each other.

length in order to exit this region (robot i needs to travel distance $l_j + L_i$ in order to exit this region). It follows that in the coordination space, the obstacle region is a rectangular region of length $l_j + L_i$ along axis i and $l_i + L_j$ along axis j (see the right drawing of Figure 2.5).

Example 2.1.3 (Two robots along the same straight path)**.** Finally, consider two robots of length L traveling along the same straight paths as depicted in Figure 2.6 and assume that the same origin is used for the curvilinear coordinate of both robots. There are two options: either robot i follows robot j and collision avoidance requires $x_j \geq x_i + L$, or robot j follows robot i and collision avoidance requires $x_i \geq x_j + L$. Hence, the collision avoidance requirement including both cases is: $|x_i - x_j| \geq L$, and the obstacle region should be the band $\{x \in \chi : |x_i - x_j| < L\}$. However, we do not aim to model the collisions in an infinite spatial region. Hence, the band is truncated as depicted in Figure 2.6.

2.2 The priority relation

2.2.1 The completed obstacle region

This subsection shows that the intuitive notion of "assigning priorities" is equivalent to a completion of the obstacle region. It is indeed equivalent to consider as forbidden configurations both collision configurations and configurations that do not respect the assigned priorities, resulting in a completed obstacle region.

Let $\chi^{\text{obs}}_{i \succ j}$ and $\chi^{\text{free}}_{i \succ j}$ denote the subsets of χ defined below:

$$\chi^{\text{obs}}_{i \succ j} := \chi^{\text{obs}}_{ij} - \mathbb{R}_+ \mathbf{e}_i + \mathbb{R}_+ \mathbf{e}_j \tag{2.12}$$

$$\chi^{\text{free}}_{i \succ j} := \chi \setminus \chi^{\text{obs}}_{i \succ j} \tag{2.13}$$

We also define $\kappa_{i \succ j} \subset \mathbb{R}^2$ as follows:

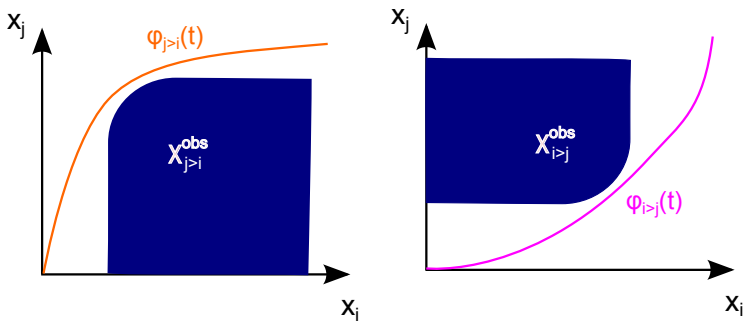


Figure 2.7: Projection of the completed collision cylinders $\chi^{\text{obs}}_{i \succ j}$ and $\chi^{\text{obs}}_{j \succ i}$.

$$\kappa_{i \succ j} := \kappa_{ij} + \mathbb{R}_- \times \mathbb{R}_+ \tag{2.14}$$

which is the cross-section of $\chi^{\text{obs}}_{i \succ j}$, i.e.,

$$\chi^{\text{obs}}_{i \succ j} = \{x \in \chi : (x_i, x_j) \in \kappa_{i \succ j}\} \tag{2.15}$$

Figure 2.7 displays the sets $\chi^{\text{obs}}_{i\succ j}$ and $\chi^{\text{obs}}_{j\succ i}$. The rationale behind the definition of these sets is that as a feasible path is non-decreasing, it necessarily lies below or above each collision cylinder as depicted in Figure 2.7. This reflects the intuitive notion of priority at intersections. Deciding on which side to pass with respect to each collision cylinder is equivalent to deciding the relative order of robots to go through the intersection. In the sequel, we are going to prove that the definition of the sets $\chi^{\text{obs}}_{i\succ j}$ enables to define rigorously the so-called priority relation induced by a feasible path. We start with some geometric properties that will be used in the proofs of the presented results.

Property 2.2.1 (Geometric invariances of $\chi^{\text{obs}}_{i\succ j}$ and $\chi^{\text{free}}_{i\succ j}$ illustrated in Figure 2.8)**.** *For all $i, j \in \mathcal{R}$, the following identities hold:*

$$\chi^{\text{obs}}_{i\succ j} - \mathbb{R}_+\mathbf{e}_i + \mathbb{R}_+\mathbf{e}_j = \chi^{\text{obs}}_{i\succ j} \tag{2.16}$$
$$\chi^{\text{free}}_{i\succ j} + \mathbb{R}_+\mathbf{e}_i - \mathbb{R}_+\mathbf{e}_j = \chi^{\text{free}}_{i\succ j} \tag{2.17}$$

Property 2.2.2 (Invariance through min and max operators illustrated in Figure 2.8)**.** *Given $x, y \in \chi$, for all $i, j \in \mathcal{R}$, the following implications hold:*

$$x, y \in \chi^{\text{free}}_{i\succ j} \Rightarrow \max\{x, y\} \in \chi^{\text{free}}_{i\succ j} \tag{2.18}$$
$$x, y \in \chi^{\text{free}}_{i\succ j} \Rightarrow \min\{x, y\} \in \chi^{\text{free}}_{i\succ j} \tag{2.19}$$

Property 2.2.3 (Illustrated in Figure 2.9)**.** *For all $i, j \in \mathcal{R}$ and $y \in \chi^{\text{obs}}_{ij}$, we have:*

$$\{x \in \chi : x_i = y_i\} \subset \left(\chi^{\text{obs}}_{i\succ j} \cup \chi^{\text{obs}}_{j\succ i}\right) \tag{2.20}$$

Property 2.2.4 (Illustrated in Figure 2.9)**.** *For all $i, j \in \mathcal{R}$, given $x^1 \in \chi^{\text{obs}}_{j\succ i}$ and $x^2 \in \chi^{\text{obs}}_{i\succ j}$, we have:*

$$\left\{x \in \chi : x^1_i \le x_i \le x^2_i\right\} \subset \left(\chi^{\text{obs}}_{i\succ j} \cup \chi^{\text{obs}}_{j\succ i}\right) \tag{2.21}$$

Proof of Property 2.2.1. Take $i, j \in \mathcal{R}$. By simple manipulations,

$$\chi^{\text{obs}}_{i\succ j} - \mathbb{R}_+\mathbf{e}_i + \mathbb{R}_+\mathbf{e}_j = (\chi^{\text{obs}}_{ij} - \mathbb{R}_+\mathbf{e}_i + \mathbb{R}_+\mathbf{e}_j) - \mathbb{R}_+\mathbf{e}_i + \mathbb{R}_+\mathbf{e}_j = \chi^{\text{obs}}_{ij} - (\mathbb{R}_+ + \mathbb{R}_+)\mathbf{e}_i + (\mathbb{R}_+ + \mathbb{R}_+)\mathbf{e}_j = \chi^{\text{obs}}_{ij} - \mathbb{R}_+\mathbf{e}_i + \mathbb{R}_+\mathbf{e}_j = \chi^{\text{obs}}_{i\succ j} \tag{2.22}$$

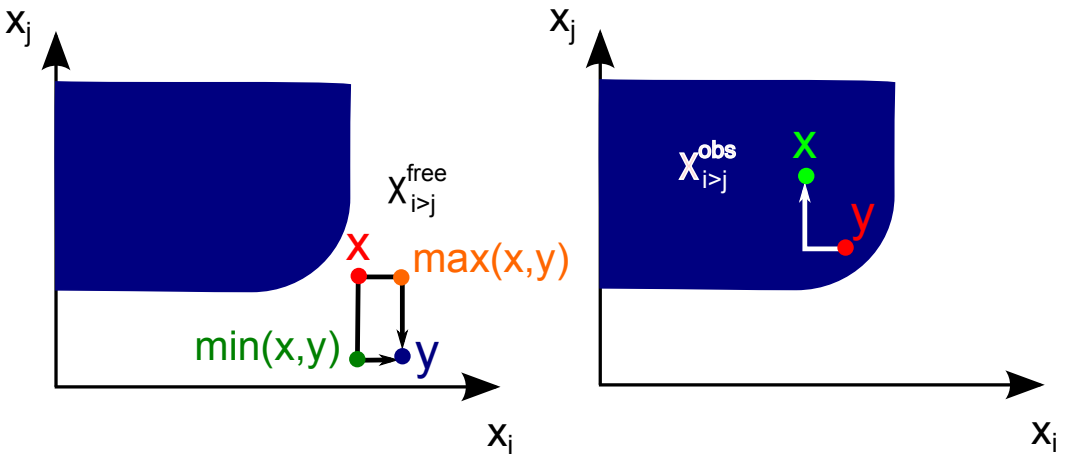


Figure 2.8: Illustration of Properties 2.2.1 and 2.2.2.

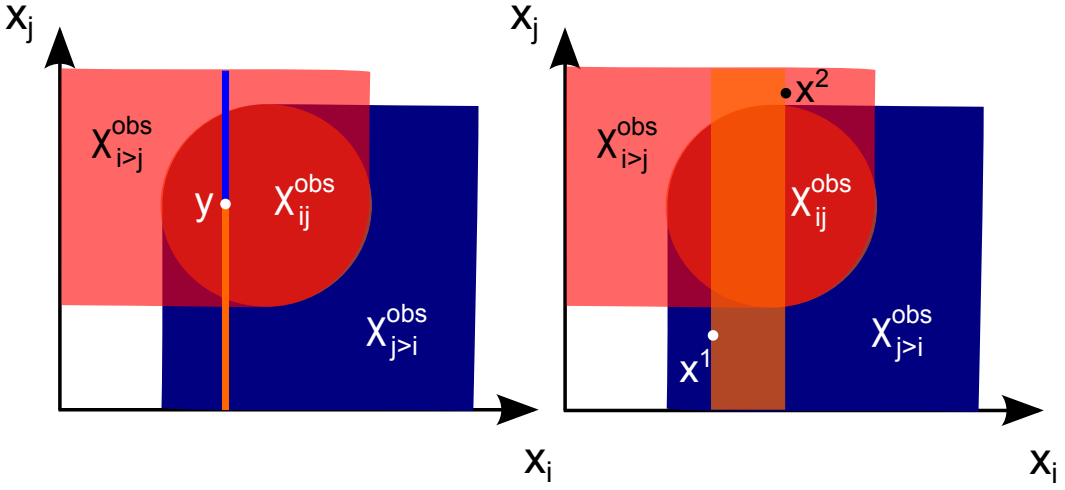


Figure 2.9: Illustration of Properties 2.2.3 and 2.2.4.

We have obtained (2.16). Moreover, using the latter result, we have:

$$x \in \chi^{\text{free}}_{i\succ j} \Leftrightarrow x \notin \chi^{\text{obs}}_{i\succ j} \Leftrightarrow x \notin \left(\chi^{\text{obs}}_{i\succ j} - \mathbb{R}_+\mathbf{e}_i + \mathbb{R}_+\mathbf{e}_j\right) \tag{2.23}$$

Hence, we have:

$$\begin{aligned} x \in \chi^{\text{free}}_{i\succ j} \Leftrightarrow \forall \alpha, \beta \geq 0, x \notin \left(\chi^{\text{obs}}_{i\succ j} - \alpha\mathbf{e}_i + \beta\mathbf{e}_j\right) \Leftrightarrow \\ \forall \alpha, \beta \geq 0, (x + \alpha\mathbf{e}_i - \beta\mathbf{e}_j) \notin \chi^{\text{obs}}_{i\succ j} \Leftrightarrow \forall \alpha, \beta \geq 0, (x + \alpha\mathbf{e}_i - \beta\mathbf{e}_j) \in \chi^{\text{free}}_{i\succ j} \end{aligned} \tag{2.24}$$

We have obtained (2.17). □

Proof of (2.18) *in Property 2.2.2.* Take $i, j \in \mathcal{R}$, $x, y \in \chi^{\text{free}}_{i\succ j}$ and let $z := \max\{x, y\}$. By definition of max, there are four options:

- $z_i = x_i$ and $z_j = x_j$: in this case, $(z_i, z_j) = (x_i, x_j) \notin \kappa_{i\succ j}$, so that $z \in \chi^{\text{free}}_{i\succ j}$.
- $z_i = y_i$ and $z_j = y_j$: this is the symmetric case and $y \in \chi^{\text{free}}_{i\succ j}$ implies that $z \in \chi^{\text{free}}_{i\succ j}$.
- $z_i = x_i \geq y_i$ and $z_j = y_j$: in this case, we have $z_i \geq y_i$ and $z_j = y_j$. By Property 2.2.1, $y \in \chi^{\text{free}}_{i\succ j}$ implies that $z \in \chi^{\text{free}}_{i\succ j}$
- $z_i = y_i \geq x_i$ and $z_j = x_j$: this is the symmetric case and $x \in \chi^{\text{free}}_{i\succ j}$ implies that $z \in \chi^{\text{free}}_{i\succ j}$

□

Proof of (2.19) *in Property 2.2.2.* Take $i, j \in \mathcal{R}$, $x, y \in \chi^{\text{free}}_{i\succ j}$ and let $z := \min\{x, y\}$. By definition of min, there are four options:

- $z_i = x_i$ and $z_j = x_j$: in this case, $(z_i, z_j) = (x_i, x_j) \notin \kappa_{i\succ j}$, so that $z \in \chi^{\text{free}}_{i\succ j}$.
- $z_i = y_i$ and $z_j = y_j$: this is the symmetric case and $y \in \chi^{\text{free}}_{i\succ j}$ implies that $z \in \chi^{\text{free}}_{i\succ j}$.
- $z_i = x_i$ and $z_j = y_j \leq x_j$: in this case, we have $z_i = x_i$ and $z_j \leq x_j$. By Property 2.2.1, $x \in \chi^{\text{free}}_{i\succ j}$ implies that $z \in \chi^{\text{free}}_{i\succ j}$
- $z_i = y_i$ and $z_j = x_j \leq y_j$: this is the symmetric case and $y \in \chi^{\text{free}}_{i\succ j}$ implies that $z \in \chi^{\text{free}}_{i\succ j}$

□

Proof of Property 2.2.3. Take $i, j \in \mathcal{R}$, $y \in \chi^{\text{obs}}_{ij}$ and $x \in \{x \in \chi : x_i = y_i\}$. Either $x_j \geq y_j$ and by Property 2.2.1 $y \in \chi^{\text{obs}}_{ij} \subset \chi^{\text{obs}}_{i\succ j}$ implies that $x \in \chi^{\text{obs}}_{i\succ j}$; or, $x_j \leq y_j$ and by Property 2.2.1 $y \in \chi^{\text{obs}}_{ij} \subset \chi^{\text{obs}}_{j\succ i}$ implies that $x \in \chi^{\text{obs}}_{j\succ i}$. In both cases, we have $x \in \chi^{\text{obs}}_{i\succ j} \cup \chi^{\text{obs}}_{j\succ i}$. □

Proof of Property 2.2.4. Take $x^1 \in \chi^{\text{obs}}_{j\succ i}$, $x^2 \in \chi^{\text{obs}}_{i\succ j}$, and $x \in \chi$ satisfying $x^1_i \leq x_i \leq x^2_i$. As $x^1 \in \chi^{\text{obs}}_{j\succ i}$ (which is non-empty open and lower-bounded along axis i with the same bound as χ^{obs}_{ij}), $x^1_i > \inf\{y_i : y \in \chi^{\text{obs}}_{ji}\}$ and as $x^2 \in \chi^{\text{obs}}_{i\succ j}$ (which is non-empty open and upper-bounded along axis i with the same bound as χ^{obs}_{ij}), $x^2_i < \sup\{y_i : y \in \chi^{\text{obs}}_{ij}\}$. Hence, we obtain:

$$\inf\{y_i : y \in \chi^{\text{obs}}_{ij}\} < x_i < \sup\{y_i : y \in \chi^{\text{obs}}_{ij}\} \tag{2.25}$$

As a consequence, there exists $x^0 \in \chi^{\text{obs}}_{ij}$ such that $x^0_i = x_i$. By Property 2.2.3, we obtain $x \in \left(\chi^{\text{obs}}_{i\succ j} \cup \chi^{\text{obs}}_{j\succ i}\right)$. □

2.2.2 The priority relation

The definition of the completed obstacle region enables to easily define a priority relation for feasible paths. The fact that a feasible path necessarily and exclusively lies on one side or on the other side of each collision cylinder χ^{obs}_{ij} is indeed equivalent to intersect, necessarily and exclusively, one of the completed cylinders $\chi^{\text{obs}}_{i\succ j}$, or $\chi^{\text{obs}}_{j\succ i}$.

Definition 2.2.1 (Priority relation)**.** *The priority relation $\succ$ is a binary relation on the set of robots $\mathcal{R}$. For all $i, j \in \mathcal{R}$, $i \succ j$ if $\text{Im}(\varphi) \cap \chi^{\text{obs}}_{j\succ i} \neq \emptyset$.*

We say $\succ$ is the priority relation induced by φ. The theorem below shows that the relation $\succ$ satisfies basic properties that one can expect from a "priority relation". More precisely, $\succ$ does not define a priority relation between two robots that cannot collide ($\chi^{\text{obs}}_{ij} = \emptyset$) and if two robots can potentially collide, a priority relation exists and we have $i \succ j$ or $j \succ i$ exclusively, i.e., if robot i has priority over robot j then robot j does not have priority over robot i.

Theorem 2.2.1 (Priority relation properties)**.** *Let $\varphi \in \Phi(\chi^{\text{free}})$ denote a feasible path and $\succ$ the priority relation induced by φ. For all $i, j \in \mathcal{R}$ such that $\chi^{\text{obs}}_{ij} \neq \emptyset$, we have necessarily and exclusively $i \succ j$ or $j \succ i$. For all $i, j \in \mathcal{R}$ such that $\chi^{\text{obs}}_{ij} = \emptyset$, we have $i \not\succ j$.*

Note that the first statement of the above theorem can be formulated synthetically as: the binary relation $\succ$ is asymmetric. To prove Theorem 2.2.1, we start with the following lemma illustrated in Figures 2.10 and 2.11 and proved in Appendix A.2:

Lemma 2.2.1 (South-West and North-East completion [97])**.** *For all feasible paths $\varphi \in \Phi(\chi^{\text{free}})$,*

$$\forall i, j \in \mathcal{R}, \text{Im}(\varphi) \cap \left(\chi^{\text{obs}}_{i\succ j} \cap \chi^{\text{obs}}_{j\succ i}\right) = \emptyset \tag{2.26}$$

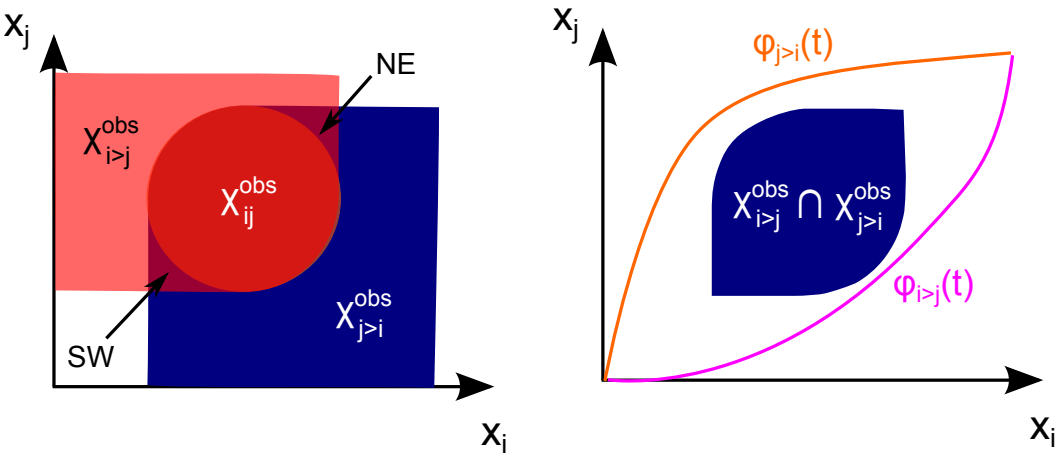


Figure 2.10: Illustration of Lemma 2.2.1. Both $\varphi_{i\succ j}$ and $\varphi_{j\succ i}$ are collision-free with regards to $\chi^{\text{obs}}_{i\succ j} \cap \chi^{\text{obs}}_{j\succ i}$. Compared to χ^{obs}_{ij}, $\chi^{\text{obs}}_{i\succ j} \cap \chi^{\text{obs}}_{j\succ i}$ additionally contains the south-west (SW) region of the obstacle region and the north-east (NE) region of the obstacle region. Feasible paths do not go through the south-west region, as it necessarily leads to a "deadlock" between robots i and j. The north-east region cannot be reached by a feasible (non-decreasing) path.

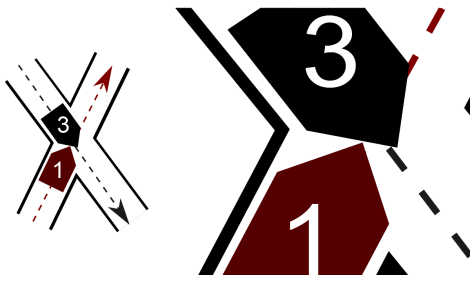


Figure 2.11: Two robots at a deadlock configuration in the south-west region.

Note that it was already noticed in [97] that south-west completion enables to avoid deadlocks in two-robot systems.

Proof of Theorem 2.2.1. Take a feasible path $\varphi \in \Phi(\chi^{\text{free}})$ and let $\succ$ denote the priority relation induced by φ. Take $i, j \in \mathcal{R}$ such that $\chi_{ij}^{\text{obs}} = \emptyset$. Then, we have $\chi_{j\succ i}^{\text{obs}} = \emptyset$, so that $\text{Im}\,(\varphi) \cap \chi_{j\succ i}^{\text{obs}} = \emptyset$, that is $i \not\succ j$. Take $i, j \in \mathcal{R}$ such that $\chi_{ij}^{\text{obs}} \neq \emptyset$ and take $y \in \chi_{ij}^{\text{obs}}$. Remember that we have $(\varphi(0) - \mathbb{R}_+^n) \subset \chi^{\text{free}}$ and $(\varphi(1) + \mathbb{R}_+^n) \subset \chi^{\text{free}}$. As a consequence, there are two options as depicted in Figure 2.12:

(a) either $y \in \text{Im}\,(\varphi) - \mathbb{R}_+\mathbf{e}_i + \mathbb{R}_+\mathbf{e}_j$: it implies that $\text{Im}\,(\varphi) \cap \chi_{j\succ i}^{\text{obs}} \neq \emptyset$;

(b) or $y \in \text{Im}\,(\varphi) - \mathbb{R}_+\mathbf{e}_j + \mathbb{R}_+\mathbf{e}_i$: it implies that $\text{Im}\,(\varphi) \cap \chi_{i\succ j}^{\text{obs}} \neq \emptyset$.

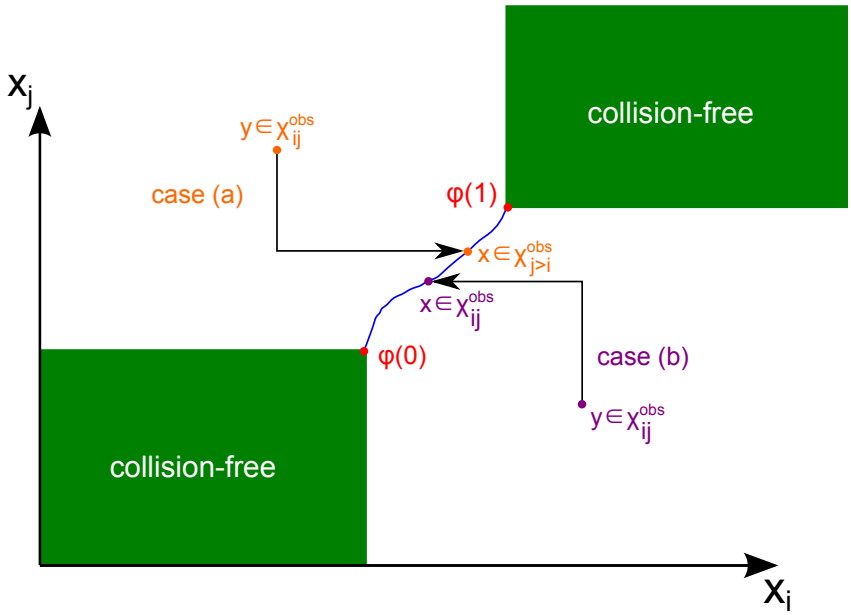


Figure 2.12: The two cases that appear to prove that any path $\varphi \in \Phi(\chi^{\text{free}})$ necessarily intersects $\chi_{i\succ j}^{\text{obs}}$ or $\chi_{j\succ i}^{\text{obs}}$ provided $\chi_{ij}^{\text{obs}} \neq \emptyset$.

Hence, a feasible path necessarily intersects $\chi_{i\succ j}^{\text{obs}}$ or $\chi_{j\succ i}^{\text{obs}}$, so we have necessarily $i \succ j$ or $j \succ i$.

Now, we will prove that it is exclusive by contradiction. Take a feasible path φ and assume that for some $t^1 \in [0,1]$, $\varphi(t^1) \in \chi_{i\succ j}^{\text{obs}}$ and for some $t^2 \in [0,1]$, $\varphi(t^2) \in \chi_{j\succ i}^{\text{obs}}$. Assume arbitrarily that $t^1 \leq t^2$ (otherwise, exchange the roles of i and j), which implies that $\varphi(t^1) \leq \varphi(t^2)$. Consider the subset of χ defined below:

$$K := \left\{ x \in \chi : x_i^1 \leq x_i \leq x_i^2 \text{ and } x_j^1 \leq x_j \leq x_j^2 \right\} \tag{2.27}$$

By Property 2.2.4, we have:

$$K \subset \left(\chi_{i\succ j}^{\text{obs}} \cup \chi_{j\succ i}^{\text{obs}} \right) \tag{2.28}$$

As φ is non-decreasing, for all $t \in [t^1, t^2]$, $\varphi(t) \in K$. If $\varphi(t) \in \chi^{\text{obs}}_{i\succ j} \cap \chi^{\text{obs}}_{j\succ i}$ for some $t \in [t^1, t^2]$, φ would not be feasible by Lemma 2.2.1. Hence, we have:

$$\varphi(t^1) \in \chi^{\text{obs}}_{i\succ j} \setminus \chi^{\text{obs}}_{j\succ i} \tag{2.29}$$

$$\varphi(t^2) \in \chi^{\text{obs}}_{j\succ i} \setminus \chi^{\text{obs}}_{i\succ j} \tag{2.30}$$

and for all $t \in [t^1, t^2]$,

$$\varphi(t) \in \left(\chi^{\text{obs}}_{i\succ j} \setminus \chi^{\text{obs}}_{j\succ i}\right) \cup \left(\chi^{\text{obs}}_{j\succ i} \setminus \chi^{\text{obs}}_{i\succ j}\right) \tag{2.31}$$

As $\left(\chi^{\text{obs}}_{i\succ j} \setminus \chi^{\text{obs}}_{j\succ i}\right) \cap \left(\chi^{\text{obs}}_{j\succ i} \setminus \chi^{\text{obs}}_{i\succ j}\right) = \emptyset$, by continuity of φ (see Lemma A.1.1 in Appendix A.1), there exists some $t^0 \in [t^1, t^2]$ such that:

$$\varphi(t^0) \in \partial\left(\chi^{\text{obs}}_{i\succ j} \setminus \chi^{\text{obs}}_{j\succ i}\right) \cap \partial\left(\chi^{\text{obs}}_{j\succ i} \setminus \chi^{\text{obs}}_{i\succ j}\right) \tag{2.32}$$

As $\chi^{\text{obs}}_{i\succ j}$ and $\chi^{\text{obs}}_{j\succ i}$ are open, by Lemma A.1.2 (see Appendix A.1), we have $\chi^{\text{obs}}_{j\succ i} \cap \partial(\chi^{\text{obs}}_{i\succ j} \setminus \chi^{\text{obs}}_{j\succ i}) = \emptyset$ and $\chi^{\text{obs}}_{i\succ j} \cap \partial(\chi^{\text{obs}}_{j\succ i} \setminus \chi^{\text{obs}}_{i\succ j}) = \emptyset$. Hence, we obtain:

$$\partial\left(\chi^{\text{obs}}_{i\succ j} \setminus \chi^{\text{obs}}_{j\succ i}\right) \cap \partial\left(\chi^{\text{obs}}_{j\succ i} \setminus \chi^{\text{obs}}_{i\succ j}\right) \cap \left(\chi^{\text{obs}}_{i\succ j} \cup \chi^{\text{obs}}_{j\succ i}\right) = \emptyset \tag{2.33}$$

Equations (2.31) and (2.32) are therefore absurd as disjoint sets have no element in common. □

2.2.3 The priority graph

As any binary relation, the priority relation admits a graph representation.

Definition 2.2.2 (Priority graph)**.** *The priority graph induced by a feasible path φ is the oriented graph G whose vertices are $V(G) := \mathcal{R}$ and such that there is an edge from i to j if $i \succ j$ where $\succ$ denotes the priority relation induced by φ. We write $(i, j) \in E(G)$ where $E(G)$ denotes the edge set of the priority graph.*

Two representations of the priority graph are depicted in Figure 2.13. We let Γ denote the application that returns the priority graph $\Gamma(\varphi)$ induced by a feasible path $\varphi \in \Phi(\chi^{\text{free}})$. $\Gamma(\varphi)$ is the graph of the priority relation $\succ$ induced by φ. Theorem 2.2.1 can be rewritten as follows:

$$\forall \varphi \in \Phi(\chi^{\text{free}}), \Gamma(\varphi) \in \mathcal{G} \tag{2.34}$$

where $\mathcal{G}$ is the set of oriented graphs G with vertices $V(G) := \mathcal{R}$, whose edge set $E(G)$ satisfies:

$$\forall i, j \in \mathcal{R}, \quad (i, j) \in E(G) \Leftrightarrow \begin{cases} \chi^{\text{obs}}_{ij} & \neq & \emptyset \\ (j, i) & \notin & E(G) \end{cases} \tag{2.35}$$

We say a graph G is a priority graph if $G \in \mathcal{G}$. It is natural as a graph $G \in \mathcal{G}$ defines a binary relation between robots whose paths intersect, i.e., it defines a

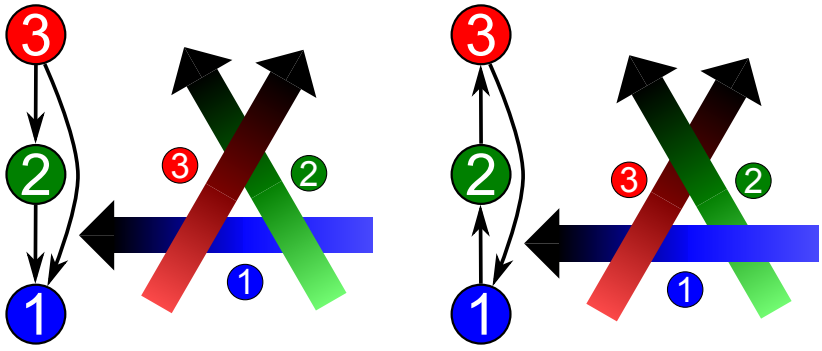


Figure 2.13: Two representations of priority relations. In each drawing relation is represented in two ways: as a complete oriented graph, where orientation yields the priority; and as trajectories over time, foreground being first, background later. The left drawing represents a relation that is an order (even a total order). The right drawing shows a relation that is not an order.

priority between all and only robots that need to coordinate. Then, a natural question is: given a priority graph G, does a feasible path exist whose induced priority graph is G ? Let $\Gamma^{-1}(G)$ denote the set of feasible paths whose induced priority graph is G. The question can then be rephrased as: given a priority graph G, do we have $\Gamma^{-1}(G) \neq \emptyset$? If there exists some path $\varphi \in \Gamma^{-1}(G)$, φ should be collision-free with regards to each completed cylinder $\chi^{\text{obs}}_{i \succ j}$ for all $(i,j) \in E(G)$. Hence, it is natural to define the completed obstacle region and the collision-free region with regards to a given priority graph $G \in \mathcal{G}$ as follows:

$$\chi^{\text{obs}}_G := \bigcup_{(i,j)\in E(G)} \chi^{\text{obs}}_{i\succ j} \tag{2.36}$$

$$\chi^{\text{free}}_G := \chi \setminus \chi^{\text{obs}}_G \tag{2.37}$$

$\{\chi^{\text{obs}}_G, \chi^{\text{free}}_G\}$ form a partition of χ. By construction, we have $\chi^{\text{obs}} \subset \chi^{\text{obs}}_G$: respecting the assigned priorities requires remaining in a more constrained space. For all feasible paths $\varphi \in \Phi(\chi^{\text{free}})$, we have the equivalences:

$$\varphi \in \Gamma^{-1}(G) \Leftrightarrow \text{Im}\,(\varphi) \subset \chi^{\text{free}}_G \Leftrightarrow \text{Im}\,(\varphi) \cap \chi^{\text{obs}}_G = \emptyset \Leftrightarrow \varphi \in \Phi(\chi^{\text{free}}_G) \tag{2.38}$$

It results that the set of feasible paths is the union of the sets of feasible paths respecting a certain priority graph over all possible priority graphs:

$$\Phi(\chi^{\text{free}}) = \bigcup_{G\in\mathcal{G}} \Phi(\chi^{\text{free}}_G) \tag{2.39}$$

However, $\left\{\Phi(\chi^{\text{free}}_G) : G \in \mathcal{G}\right\}$ do not form a partition of $\Phi(\chi^{\text{free}})$ as some $\Phi(\chi^{\text{free}}_G)$ may be empty.

The next chapter studies the coordination under assigned priorities, i.e., when the obstacle region is completed with configurations not respecting the assigned

priorities, forming the completed obstacle region χ_G^{obs}. In Section 3.1, we will see that each non-empty set of feasible paths respecting a certain priority graph G, i.e., each non-empty $\Phi(\chi_G^{\text{free}})$, is a homotopy class of feasible paths continuously deformable into each other. Section 3.2 provides a necessary and sufficient condition on G for $\Phi(\chi_G^{\text{free}})$ not to be empty, that is a necessary and sufficient condition on priorities to guarantee that respecting these priorities, all robots can eventually go through the intersection (no deadlock).

Chapter 3

The coordination space under assigned priorities

Contents

3.1 Priorities: a homotopy invariant 37
3.1.1 Homotopy classes 37
3.1.2 Invariance of the priority graph 40
3.2 Feasible priority graphs 43
3.2.1 Sufficient condition for priorities feasibility 45
3.2.2 Safety margin 47
3.2.3 Characterization of feasible priorities 48
3.2.4 Absence of deadlocks 54

Preliminaries of this work can be found in our conference paper [48].

Sketch of the chapter The first section of the present chapter demonstrates that respecting assigned priorities does not require robots to follow a particular feasible path in the coordination space. However, the path described by robots in the coordination space needs to remain in a quite large homotopy class of feasible paths continuously deformable into each other. This homotopy class is uniquely encoded by the priority graph. The second section proves that deadlock avoidance can be guaranteed at the priority assignment phase. Either priorities are "feasible" and ensure all robots will eventually go through the intersection provided they respect the assigned priorities; or, the multi robot system will inevitably reach a deadlock configuration.

3.1 Priorities: a homotopy invariant

3.1.1 Homotopy classes

$\Phi(\chi)$ is equipped with the topology of pointwise convergence and the notion of homotopic feasible paths is defined as follows.

Definition 3.1.1 (Homotopic paths)**.** *Given two feasible paths φ^1 and φ^2, φ^1 is homotopic to φ^2 if there exists a continuous application H defined on $[0,1]$ such that $H(0) = \varphi^1$, $H(1) = \varphi^2$ and for all $\alpha \in [0,1]$, the path $H(\alpha)$ is a feasible path.*

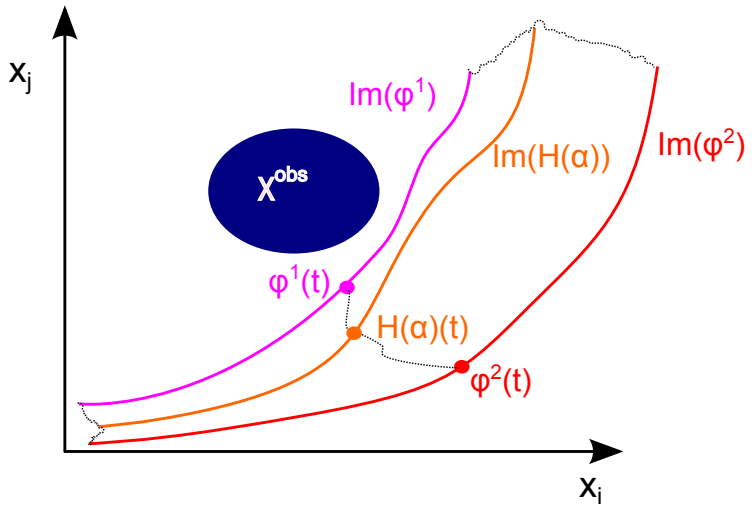


Figure 3.1: Two homotopic paths. As both paths lie below the obstacle region, they can be continuously transformed into each other remaining collision-free along the transformation.

We write $\varphi^1 \sim \varphi^2$. Said differently, two feasible paths are homotopic if they can be continuously transformed into each other remaining feasible (in particular collision-free and non-decreasing) along the transformation as depicted in Figure 3.1. Homotopy defines an equivalence relation on feasible paths:

Property 3.1.1 (Homotopy: an equivalence relation)**.** *The homotopy relation $\sim$ is an equivalence relation on $\Phi(\chi^{\text{free}})$.*

Proof. We have to prove that $\sim$ is an equivalence relation, i.e., that it is (a) reflexive, (b) symmetric and (c) transitive.

(a) Take a feasible path $\varphi \in \Phi(\chi^{\text{free}})$ and consider the constant application $H : \alpha \in [0,1] \mapsto \varphi$. $H(0) = \varphi$, $H(1) = \varphi$ and for all $\alpha \in [0,1]$, the path $H(\alpha) \equiv \varphi$ is a feasible path. Hence, $\varphi \sim \varphi$ and $\sim$ is reflexive.

(b) Take feasible paths $\varphi, \psi \in \Phi(\chi^{\text{free}})$ and assume that $\varphi \sim \psi$. Then, there exists H defined on $[0,1]$ such that $H(0) = \varphi$, $H(1) = \psi$ and for all $\alpha \in [0,1]$, the path $H(\alpha)$ is a feasible path. Consider $G : \alpha \in [0,1] \mapsto H(1-\alpha)$. We have $G(0) = \psi$, $G(1) = \varphi$ and for all $\alpha \in [0,1]$, the path $G(\alpha) \equiv H(1-\alpha)$ is a feasible path. Hence, $\psi \sim \varphi$ and $\sim$ is symmetric.

(c) Take feasible paths $\varphi^1, \varphi^2, \varphi^3 \in \Phi(\chi^{\text{free}})$ and assume that $\varphi^1 \sim \varphi^2$ and $\varphi^2 \sim \varphi^3$. Then, there exists H^{12} defined on $[0,1]$ such that $H^{12}(0) = \varphi^1$, $H^{12}(1) = \varphi^2$ and for all $\alpha \in [0,1]$, the path $H^{12}(\alpha)$ is a feasible path and there exists H^{23} defined on $[0,1]$ such that $H^{23}(0) = \varphi^2$, $H^{23}(1) = \varphi^3$ and for all $\alpha \in [0,1]$, the

path $H^{23}(\alpha)$ is a feasible path. Consider H defined on $[0,1]$ as follows:

$$\forall \alpha \in [0, 1/2], H(\alpha) := H^{12}(2\alpha) \quad (3.1)$$

$$\forall \alpha \in (1/2, 1], H(\alpha) := H^{23}(2(\alpha - 1/2)) \quad (3.2)$$

H is continuous as $\lim_{\substack{\alpha \to 1/2 \\ <}} H(\alpha) = H^{12}(1) = \varphi^2$ and $\lim_{\substack{\alpha \to 1/2 \\ >}} H(\alpha) = H^{23}(0) = \varphi^2$. Moreover, $H(0) = \varphi^1$, $H(1) = \varphi^3$ and for all $\alpha \in [0,1]$, the path $H(\alpha)$ is a feasible path as it satisfies $H(\alpha) \equiv H^{12}(2\alpha)$ or $H(\alpha) \equiv H^{23}(2(\alpha - 1/2))$ which are both feasible paths. Hence, $\varphi^1 \sim \varphi^3$ and $\sim$ is transitive.

□

As a direct consequence of Property 3.1.1, we can define homotopy classes as the equivalence classes induced by this equivalence relation. Let $\mathcal{H}^{\text{free}} := \Phi(\chi^{\text{free}})/\sim$ denote the homotopy classes of feasible paths, that is the quotient set of $\Phi(\chi^{\text{free}})$ by the equivalence relation $\sim$. Homotopy classes form a partition of $\Phi(\chi^{\text{free}})$ [61].

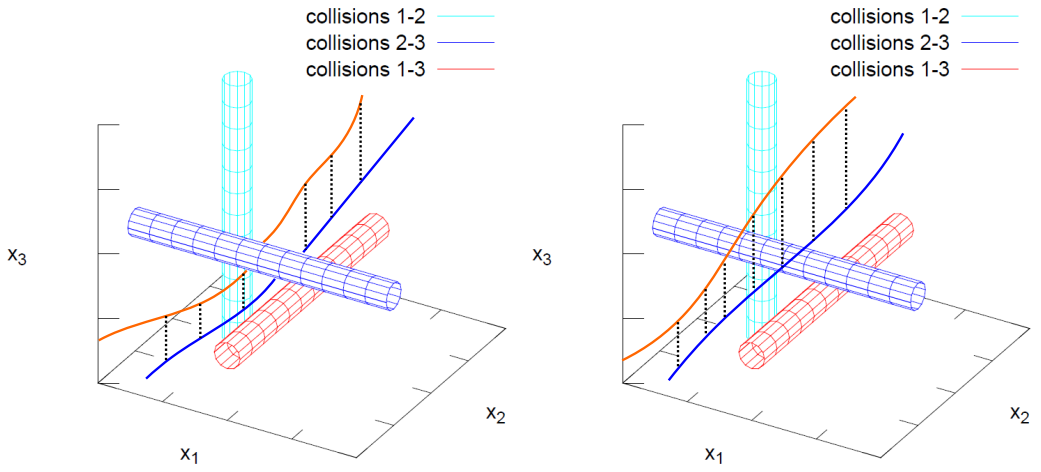


Figure 3.2: Two homotopy classes of feasible paths (and two elements of each class) in a three-dimensional coordination space.

The existence of homotopy classes of solutions to the coordination problem was already noticed, e.g., in [47] (see Figure 3.2). In [60], it is also noticed that there exist a finite number of homotopy classes of solutions to the coordination of multiple agents moving on a plane between fixed points using the concept of braids [19]. However, in that work, the geometric paths of agents is not fixed, and optimization is precisely carried out in order to find an optimal trajectory, both spatially and timely. This is not adapted for an application to the coordination of robots on roadways as the two-dimensional space is very constrained and robots have a quite low degree of freedom to choose a geometric path to go through the intersection. It thus appears much more suitable to study the homotopy classes of feasible paths in the coordination space instead of studying the homotopy classes of braids.

3.1.2 Invariance of the priority graph

If previous work already noticed the existence of homotopy classes in multi robot coordination, to our knowledge, no meaningful representative is proposed to encode homotopy classes. In the following, we present the main result of this part: priorities uniquely encode homotopy classes of feasible paths in the coordination space. The existence of a finite number of homotopy classes thus merely appears as the consequence of the finite number of possible priority graphs.

We let $\Gamma(\Phi(\chi^{\text{free}})) := \{\Gamma(\varphi) : \varphi \in \Phi(\chi^{\text{free}})\}$ denote the set of values taken by the priority graph over all feasible paths. $\Gamma(\Phi(\chi^{\text{free}}))$ is a subset of $\mathcal{G}$ containing graphs G such that there exists a feasible path $\varphi \in \Phi(\chi^{\text{free}})$ satisfying $\Gamma(\varphi) = G$. The following theorem (illustrated in Figures 3.3 and 3.4) shows that priorities and homotopy classes are strongly linked: more precisely, there is a bijective relationship between homotopy classes and "feasible priority graphs" (this term will be precisely defined in Section 3.2). We say the priority graph encodes the homotopy class.

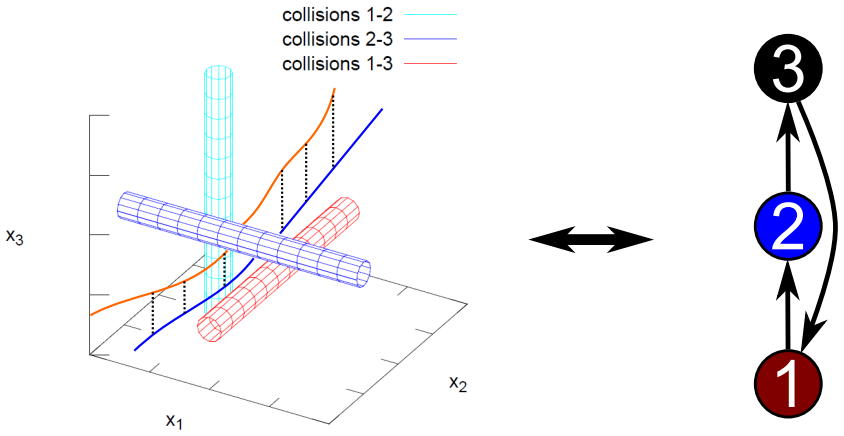


Figure 3.3: A homotopy class of feasible paths in a three-dimensional coordination space and its corresponding unique representative as a priority graph.

Theorem 3.1.1 (Invariance of the priority graph)**.** *The priority graph is an invariant of the homotopy classes of feasible paths that it is distinct for each class:* $\mathcal{H}^{\text{free}}$ *is in bijection with* $\Gamma(\Phi(\chi^{\text{free}}))$.

Proof of invariance. First we will prove that the priority graph is an invariant of the homotopy classes of feasible paths. Consider a feasible path $\varphi \in \Phi(\chi^{\text{free}})$. For all $i, j \in \mathcal{R}$, $(i,j) \in E(\Gamma(\varphi))$ if φ intersects $\chi^{\text{obs}}_{j \succ i}$ and the set $\chi^{\text{obs}}_{j \succ i}$ is open. If a feasible path φ intersects an open set, any feasible path $\psi \in \Phi(\chi^{\text{free}})$ close enough to φ (in the topology of pointwise convergence) also intersects this open set. Hence, we have:

$$\forall i, j \in \mathcal{R},\ (i,j) \in E(\Gamma(\varphi)) \Leftrightarrow (i,j) \in E(\Gamma(\psi)) \tag{3.3}$$

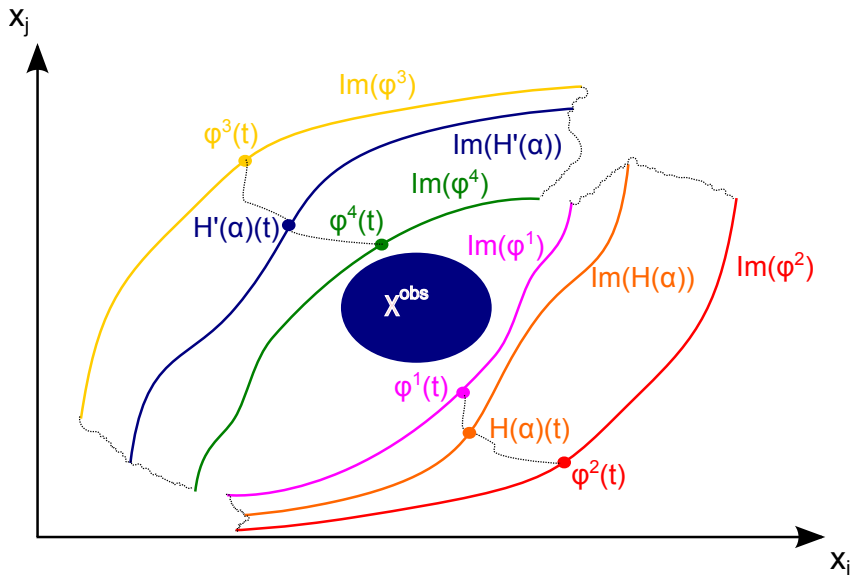


Figure 3.4: In a two-dimensional scenario ($\chi = \mathbb{R}^2$), provided $\chi_{ij}^{\text{obs}} \neq \emptyset$, there are exactly two homotopy classes: all feasible paths lying above the obstacle region form the first homotopy class and all feasible paths lying below the obstacle region form the second homotopy class.

provided ψ is close enough to φ. Therefore, Γ is continuous and since it takes discrete values, it is thus constant in homotopy classes of feasible paths. (We identify Γ with the set of applications $g_{ij} : \Phi(\chi^{\text{free}}) \to \{-1, 0, 1\}$ satisfying $g_{ij}(\varphi) = 1$ if $i \succ j$, -1 if $j \succ i$, and 0 otherwise.) In conclusion, the priority graph is an invariant of the homotopy classes of feasible paths. □

Proof of uniqueness. To prove uniqueness, consider two feasible paths φ^1 and φ^2 with the same induced priority graph G: $\varphi^1, \varphi^2 \in \Phi(\chi_G^{\text{free}})$. We have to prove that φ^1 and φ^2 are homotopic. Consider the following continuous transformation:

$$H : \alpha \in [0,1] \mapsto \min\left\{\alpha\varphi^2(1) + (1-\alpha)\varphi^1(1), \varphi^1(\bullet + \alpha), \max\left\{\varphi^1, \varphi^2\right\}\right\} \tag{3.4}$$

where by convention $\varphi^1(t + \alpha) \equiv \varphi^1(1)$ if $t + \alpha \geq 1$. Figure 3.5 illustrates the proposed transformation in the particular case where the two paths have the same endpoints.

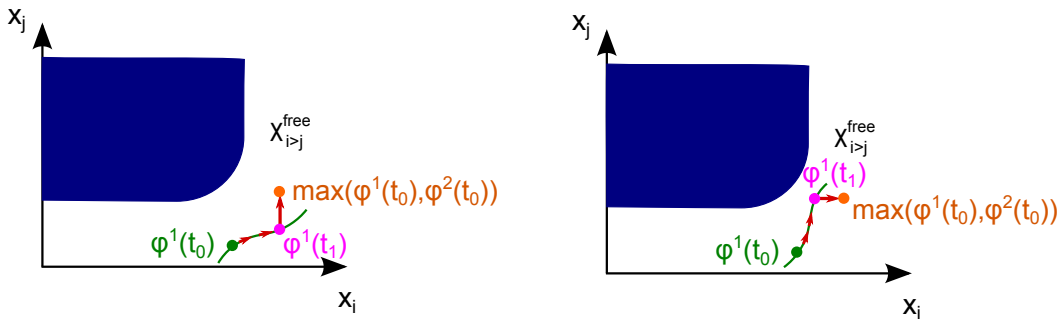


Figure 3.5: Illustration of the transformation of φ^1 into $\max(\varphi^1, \varphi^2)$. At any point of time t_0, $\max(\varphi^1(t_0), \varphi^2(t_0))$ necessarily lies on the north-east with respect to $\varphi^1(t_0)$. As a consequence, the two above cases may appear, and in each case, $\varphi^1(t_0)$ can be continuously transformed into $\max(\varphi^1(t_0), \varphi^2(t_0))$ without collision by following the red arrows.

H is continuous,

$$H(0) = \min\left\{\varphi^1(1), \varphi^1, \max\left\{\varphi^1, \varphi^2\right\}\right\} = \varphi^1 \tag{3.5}$$

$$H(1) = \min\left\{\varphi^2(1), \varphi^1(1), \max\left\{\varphi^1, \varphi^2\right\}\right\} \tag{3.6}$$

Hence, H continuously transforms φ^1 into $\min\{\varphi^2(1), \varphi^1(1), \max\{\varphi^1, \varphi^2\}\}$. Now, we prove that for all $\alpha \in [0,1]$, $H(\alpha)$ is a feasible path. We need to prove that for all $\alpha \in [0,1]$, (a) $H(\alpha)$ is continuous, (b) satisfies $(H(\alpha)(0) - \mathbb{R}^n_+) \subset \chi^{\text{free}}$ and (c) $(H(\alpha)(1) + \mathbb{R}^n_+) \subset \chi^{\text{free}}$, (d) is non-decreasing, and (e) is collision-free.

(a) $H(\alpha)$ is continuous as the result of the application of continuous operators min, max and delay on continuous paths.

(b) φ^1 and φ^2 being feasible, we have $(\varphi^1(0) - \mathbb{R}^n_+) \subset \chi^{\text{free}}_G$ and $(\varphi^2(0) - \mathbb{R}^n_+) \subset \chi^{\text{free}}_G$. Hence, we also have $(\max(\varphi^1(0), \varphi^2(0)) - \mathbb{R}^n_+) \subset \chi^{\text{free}}_G$ by Property 2.2.2, which implies that $(H(\alpha)(0) - \mathbb{R}^n_+) \subset \chi^{\text{free}}_G \subset \chi^{\text{free}}$.

(c) φ^1 and φ^2 being feasible, we have $(\varphi^1(1) + \mathbb{R}^n_+) \subset \chi^{\text{free}}_G$ and $(\varphi^2(1) + \mathbb{R}^n_+) \subset \chi^{\text{free}}_G$. Hence, we also have $(\max(\varphi^1(1), \varphi^2(1)) + \mathbb{R}^n_+) \subset \chi^{\text{free}}_G$ by Property 2.2.2, which implies that $(H(\alpha)(1) + \mathbb{R}^n_+) \subset \chi^{\text{free}}_G \subset \chi^{\text{free}}$.

(d) $H(\alpha)$ is non-decreasing as the result of the application of non-decreasing operators min and max on non-decreasing paths.

(e) Take $(i,j) \in E(G)$ and $\alpha, t \in [0,1]$. We have $\varphi^1(t+\alpha) \in \chi^{\text{free}}_{i \succ j}$ as $\varphi^1 \in \Phi(\chi^{\text{free}}_G)$ and we have also $\max\{\varphi^1(t), \varphi^2(t)\} \in \chi^{\text{free}}_{i \succ j}$ as $\varphi^1, \varphi^2 \in \Phi(\chi^{\text{free}}_G)$ and using Property 2.2.2. Moreover, $(\varphi^1(1) + \mathbb{R}^n_+) \subset \chi^{\text{free}}_{i \succ j}$ and $(\varphi^2(1) + \mathbb{R}^n_+) \subset \chi^{\text{free}}_{i \succ j}$ imply that $\alpha\varphi^2(1) + (1-\alpha)\varphi^1(1) \in \chi^{\text{free}}_{i \succ j}$ (using Property 2.2.1). By Property 2.2.2, applying the min operator on three configurations in $\chi^{\text{free}}_{i \succ j}$ returns a configuration in $\chi^{\text{free}}_{i \succ j}$. In conclusion, we have $H(\alpha)(t) \in \chi^{\text{free}}_G \subset \chi^{\text{free}}$

As a result, φ^1 is homotopic to $\min\{\varphi^2(1), \varphi^1(1), \max\{\varphi^1, \varphi^2\}\}$. As φ^1 and φ^2 have symmetric roles, φ^2 is homotopic to $\min\{\varphi^1(1), \varphi^2(1), \max\{\varphi^2, \varphi^1\}\}$, that is $\min\{\varphi^2(1), \varphi^1(1), \max\{\varphi^1, \varphi^2\}\}$. Homotopy defining an equivalence relation, φ^1 and φ^2 are homotopic. □

Proof of bijective correspondence. For each $h \in \mathcal{H}^{\text{free}}$, take an arbitrary $\varphi^h \in h$. As the priority graph is invariant in homotopy classes, we have:

$$\Gamma(\Phi(\chi^{\text{free}})) = \left\{\Gamma(\varphi^h) : h \in \mathcal{H}^{\text{free}}\right\} \tag{3.7}$$

As the priority graph $\Gamma(\varphi^h)$ is distinct for each class $h \in \mathcal{H}^{\text{free}}$, the application $\Psi : h \in \mathcal{H}^{\text{free}} \mapsto \Gamma(\varphi^h)$ is a bijection from $\mathcal{H}^{\text{free}}$ to $\Gamma(\Phi(\chi^{\text{free}}))$. In conclusion, $\mathcal{H}^{\text{free}}$ is in bijection with $\Gamma(\Phi(\chi^{\text{free}}))$. □

We have proved that all feasible paths sharing the same priorities are continuously deformable into each other. A direct consequence of the above theorem is that there exists a finite number of homotopy classes of feasible paths. When assigning the priority between each pair of robots, there is indeed two possibilities. As there is at most $n(n-1)/2$ collision pairs i, j satisfying $\chi_{ij}^{\text{obs}} \neq \emptyset$, there is at most $2^{n(n-1)/2}$ priority graphs. There is thus a finite number of homotopy classes – at most $2^{n(n-1)/2}$ – and each homotopy class of feasible paths is uniquely encoded by a priority graph $G \in \mathcal{G}$. A natural question is: does any priority graph $G \in \mathcal{G}$ encode a (non-empty) homotopy class of feasible paths ? This mathematical question is equivalent to: given assigned priorities, is it possible for robots to go through the intersection, respecting the assigned priorities ?

3.2 Feasible priority graphs

Here, we propose to give a characterization of the set of feasible priority graphs, that we define as graphs $G \in \mathcal{G}$ such that there exists a feasible path whose induced priority graph is G:

Definition 3.2.1 (Feasible priority graph)**.** *A priority graph $G \in \mathcal{G}$ is feasible if and only if $\Phi(\chi_G^{\text{free}}) \neq \emptyset$.*

Using the application Γ, the set of feasible priority graphs can be denoted as $\Gamma(\Phi(\chi^{\text{free}}))$. We start with some examples that show that the existence of a feasible path respecting given priorities is strongly related to the notion of deadlock, and we highlight the role of priority cycles in the formation of deadlocks.

Deadlock examples First of all, consider the example on the left drawing of Figure 3.6. The assigned priorities are $1 \succ 2$, $2 \succ 3$ and $3 \succ 1$. Hence, the priority graph is cyclic. It is clear that respecting the assigned priorities leads to the deadlock configuration represented in Figure 3.6. None of the robots can move without colliding. The right drawing of Figure 3.6 gives a similar example with a larger number of robots involved in the priority cycle.

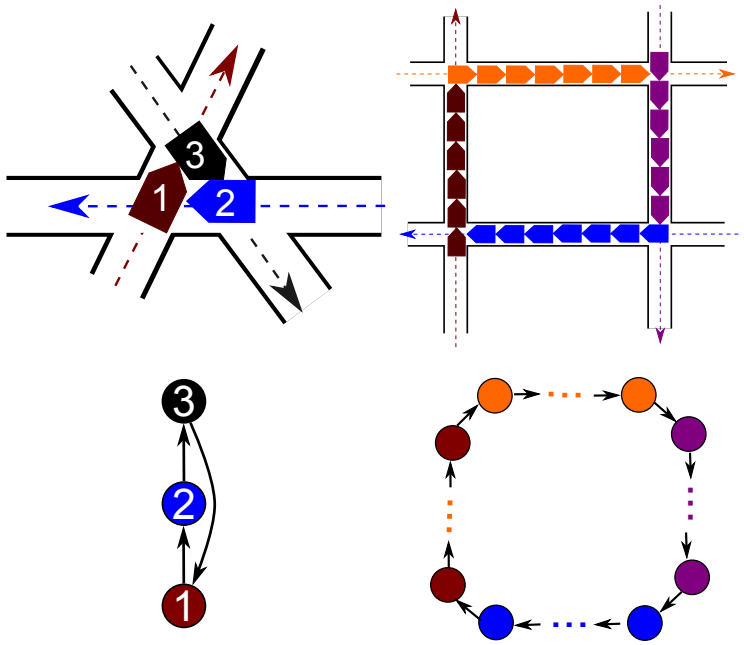


Figure 3.6: Two examples of deadlock configurations. On the left side, 3 robots are involved in the deadlock. On the right side, the deadlock is caused by a priority cycle involving much more robots. In both examples, none of the robots can move without colliding. It is a deadlock configuration.

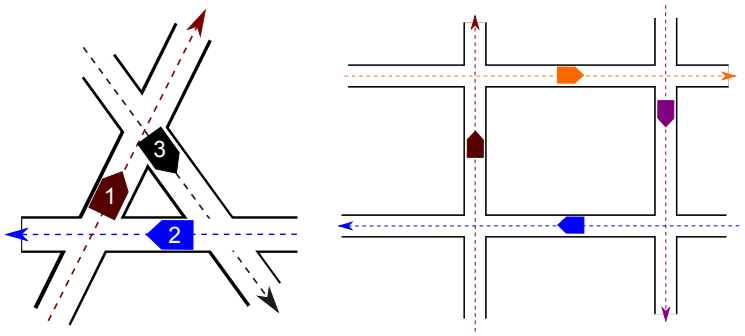


Figure 3.7: Cyclic deadlock-free examples

Cyclic deadlock-free examples According to the above example, it is clear that cycles in the priority graph have a strong role in the formation of deadlocks. Now, consider the example in the left drawing of Figure 3.7. Again, the assigned priorities, $1 \succ 3$, $2 \succ 1$ and $3 \succ 1$, are cyclic. However, it is clear that there exists a feasible path respecting the priorities and all robots will eventually go through the intersection. The right drawing of Figure 3.7 provides a similar cyclic deadlock-free example involving four robots.

The above examples justify the motivation to obtain a characterization of priority graphs such that there exists a feasible path respecting the given priorities. This characterization refines the role of cycles in the formation of deadlocks.

A singular deadlock-free priority graph Before providing such a characterization, we expose a last example where the priority graph G is feasible in that there exists a feasible path whose priority graph is G; however, all feasible paths respecting these priorities are in contact with the boundary of χ_G^{obs}. Figure 3.8 depicts such an example. It is very likely that such priorities should not be considered as feasible in practice as they require a very precise control. Note also that it is a singularity caused by the (arbitrary) openness of the obstacle region.

3.2.1 Sufficient condition for priorities feasibility

The above examples tend to indicate that no deadlock can occur under acyclic priorities. It is not surprising as deadlocks usually involve a "circular wait" [24]. In many circumstances, imposing acylic priorities is not a problematic constraint and demonstrates some benefits including deadlock avoidance (see Part III). That is why we start by providing a sufficient condition for priorities feasibility stating that acyclic priorities ensure deadlock avoidance.

Theorem 3.2.1 (Sufficient condition for priorities feasibility)**.** *All acyclic priority graphs are feasible.*

The proof of the above theorem relies on the fact that under acyclic priorities, a simple feasible path respecting the acyclic priorities can be constructed by letting robots go through the intersection one by one.

Proof. Take an acyclic priority graph $G \in \mathcal{G}$. To prove that G is feasible, we are going to exhibit a particular feasible path whose induced priority graph is G. As G is acylic, it admits a topological ordering of its nodes $\mathcal{R}$. Consider a relabeling of robots along this topological ordering, i.e., robot 1 is the maximal element of this topological ordering, ... robot i is the ith element of the topological ordering, ... and robot n is the minimal element of the topological ordering. Consider the path φ constructed as follows. $\varphi(0) := \underline{x}^{\text{obs}}$ and for all $i \in \{1 \cdots n\}$, within time interval $[(i-1)/n, i/n]$, robot i moves forward from $\underline{x}_i^{\text{obs}}$ to $\overline{x}_i^{\text{obs}}$ (for example φ_i is linear in that time interval and takes values $[\underline{x}_i^{\text{obs}}, \overline{x}_i^{\text{obs}}]$) while other robots $j \neq i$ do not

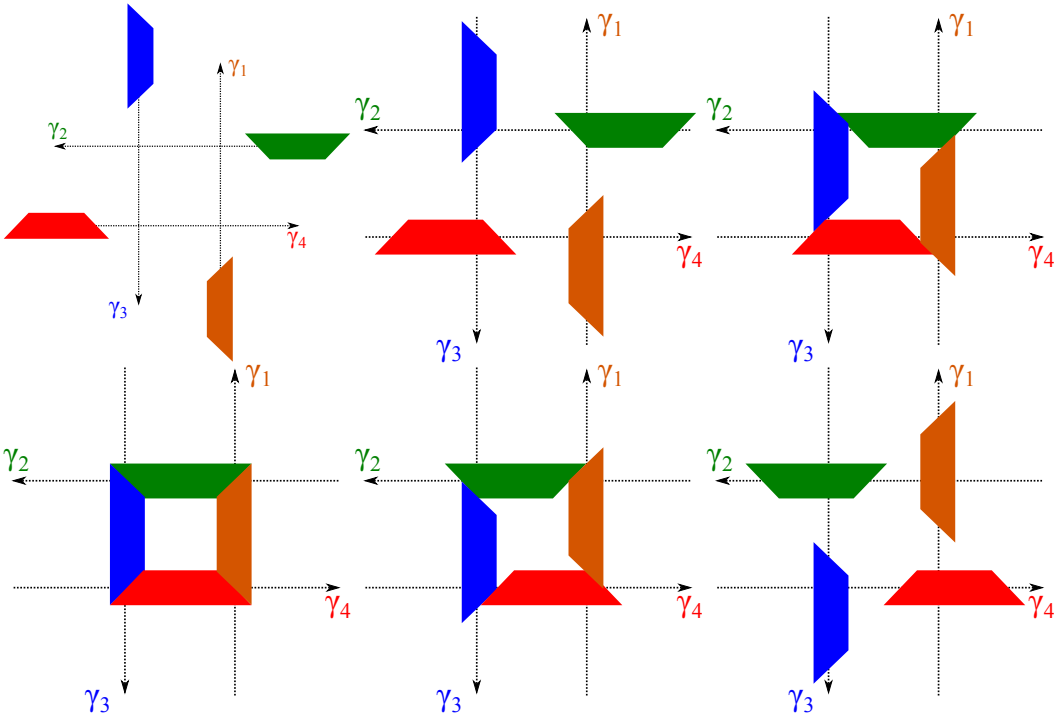


Figure 3.8: In this (quite imaginative) scenario, robots maintaining constant velocity do not collide and slide on each other to go through the intersection. It is a very singular scenario. There exists a feasible path satisfying the cyclic priorities $2 \succ 1$, $3 \succ 2$, $4 \succ 3$ *and* $1 \succ 4$, *but with absolutely no "safety margin" as robots need to slide on each other.*

move (φ_j constant in that time interval). This path is feasible and takes values in χ_G^{free}. □

Note to the reader The two following subsections intend to treat the case of cyclic yet feasible priorities. They are quite technical and the reader having no particular interest in this case can go directly to Subsection 3.2.4. It will not affect the understanding of the rest of the thesis.

3.2.2 Safety margin

As shown in the examples presented previously, only considering feasibility as a binary question is quite insufficient in practice as some priority graphs are feasible but require robots to slide on each other, i.e., to travel through very risky configurations. That is why we propose a notion of feasibility endowed with the notion of safety margin.

We say that the priority graph is feasible with a (safety) margin $r \geq 0$ if there exists a feasible path $\varphi \in \Phi(\chi_G^{\text{free}})$ keeping a distance r from the obstacle region χ_G^{obs} (in infinity norm). Given a path $\varphi \in \Phi(\chi)$, $d(\varphi, \chi_G^{\text{obs}})$ is defined as follows:

$$d(\varphi, \chi_G^{\text{obs}}) \quad := \quad \sup \left\{ r \geq 0 : \forall t \in [0,1], \forall x \in \chi_G^{\text{obs}}, \|\varphi(t) - x\|_\infty \geq r \right\} \tag{3.8}$$

When $d(\varphi, \chi_G^{\text{obs}}) \geq 0$, we say that φ is safe with regards to χ_G^{obs} with a margin $d(\varphi, \chi_G^{\text{obs}})$. The use of the distance of infinity norm makes sense since it means that a path is safe with regards to χ_G^{obs} with a margin $r \geq 0$ if robots traveling along this path with an individual precision of r will not collide (with regards to χ_G^{obs}). We have indeed the following equivalence:

$$\left[\forall x \in \chi_G^{\text{obs}},\ \|\varphi(t) - x\|_\infty \geq r \right] \Leftrightarrow \left[(\varphi(t) + [-r, r]^n) \subset \chi_G^{\text{free}} \right] \tag{3.9}$$

It is direct that the set of paths $\varphi \in \Phi(\chi_G^{\text{free}})$ satisfying $d(\varphi, \chi_G^{\text{obs}}) \geq r$ is precisely $\Phi(\chi_G^{\text{free}} \ominus [-r, r]^n)$ where $\ominus$ denotes the erosion operator, i.e.,

$$\chi_G^{\text{free}} \ominus [-r, r]^n := \left\{ x \in \chi_G^{\text{free}} : x + [-r, r]^n \subset \chi_G^{\text{free}} \right\} \tag{3.10}$$

$\chi_G^{\text{free}} \ominus [-r, r]^n$ is the erosion of χ_G^{free} with the structuring element $[-r, r]^n$ [116] (see Figure 3.9). The form of the structuring element is due to the use of the infinity norm (it is the closed ball of radius r with regards to the infinity norm). $\Phi(\chi_G^{\text{free}} \ominus [-r, r]^n)$ denotes the set of feasible paths whose priority graph is G and keeping a distance r from χ_G^{obs}. It is natural to define a safety margin associated to the priority graph G as follows:

$$\rho_G := \begin{cases} \max \left\{ r \geq 0 : \Phi(\chi_G^{\text{free}} \ominus [-r, r]^n) \neq \emptyset \right\} & \text{if } \Phi(\chi_G^{\text{free}}) \neq \emptyset \\ -\min \left\{ r > 0 : \Phi(\chi_G^{\text{free}} + [-r, r]^n) \neq \emptyset \right\} & \text{else.} \end{cases} \tag{3.11}$$

- When $\Phi(\chi_G^{\text{free}}) \neq \emptyset$, $\rho_G \in \mathbb{R}_+ \cup \{+\infty\}$ denotes the maximal distance between χ_G^{obs} and feasible paths whose priority graph is G.

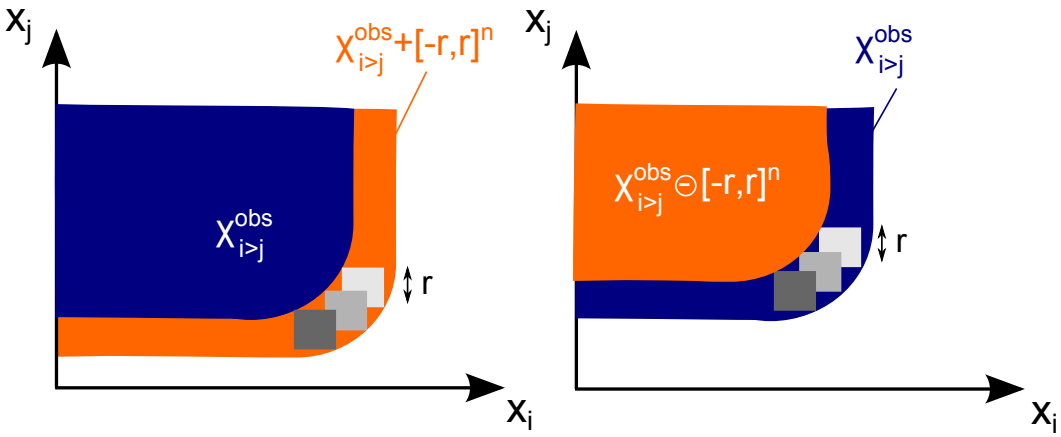


Figure 3.9: The left drawing represents the dilatation of $\chi^{\text{obs}}_{i\succ j}$ with the structuring element $[-r,r]^n$. The right drawing show the erosion of $\chi^{\text{obs}}_{i\succ j}$ with the structuring element $[-r,r]^n$.

- When $\Phi(\chi^{\text{free}}_G) = \emptyset$, $\rho_G < 0$. The value taken by $\rho_G < 0$ can be interpreted as an indicator of how far the priority graph G is from being feasible.

The use of the maximal (resp. minimal) element is justified by Lemma A.3.1 proved in Appendix A.3 that stipulates that the upper (resp. lower) bound is attained, i.e., there exists a path with maximal margin. We refer to ρ_G as the safety margin of G. This definition is coherent as a feasible priority graph has necessarily a non-negative safety margin since $\Phi(\chi^{\text{free}}_G) \neq \emptyset$.

3.2.3 Characterization of feasible priorities

Before providing a formal characterization of feasible priority graphs, a geometric interpretation is provided about why in certain circumstance cyclic priorities are not feasible. In Figure 3.10, the obstacle cylinders in the coordination space are depicted for both the cyclic deadlock-free example of Figure 3.7 and the deadlock example of Figure 3.6. The main difference is that in the deadlock case, cylinders intersect with each other. In contrast, in the deadlock-free case, cylinders do not intersect each other, there is a certain distance between each cylinder. Thanks to this distance between cylinders, the multi robot system can decide, independently for each collision cylinder, on which side to travel. On the contrary, if there is not a sufficient distance between cylinders, theses decisions are not independent. It is very clear on the right drawing of Figure 3.10 that if a feasible path lies above the obstacle cylinder χ^{obs}_{23} (the blue one) and below χ^{obs}_{13} (the red one), then it also must lie on the right relative to the obstacle cylinder χ^{obs}_{12} (the cyan one). In terms of priorities, it means that if $3 \succ 2$ and $1 \succ 3$, then we must have $1 \succ 2$, i.e., the cycle $2 \succ 1 \succ 3 \succ 2$ is forbidden.

The definition of the completed obstacle regions $\chi^{\text{obs}}_{i\succ j}$ enables to provide a very synthetic characterization of feasible priority graphs. It confirms and refines the

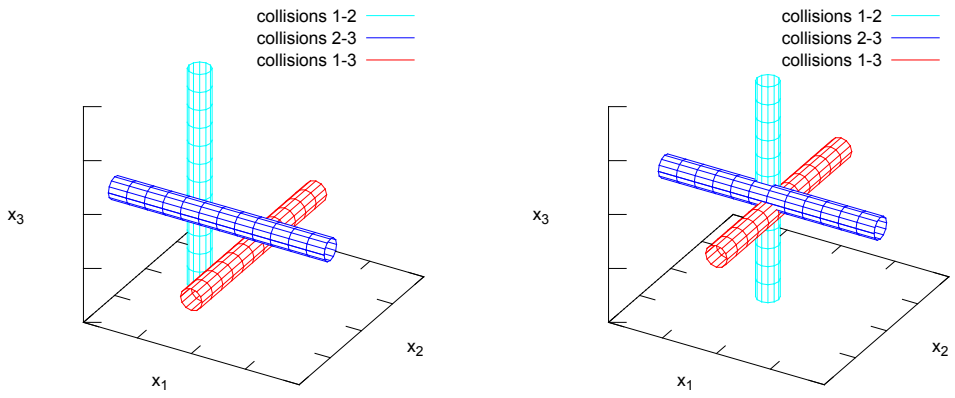


Figure 3.10: The left drawing depicts the obstacle cylinders in the coordination space for the three-robot system of Figure 3.7 in a deadlock-free configuration. The right drawing depicts the obstacle cylinders in the coordination space for the three-robot system of Figure 3.6 stuck in a deadlock.

role of priority cycles in the formation of deadlocks. In particular, Condition (3.12) gives a necessary and sufficient condition for a priority cycle to be feasible. We let cycles(G) denote the elementary cycles of a priority graph $G \in \mathcal{G}$.

Theorem 3.2.2 (Characterization of feasible priority graphs)**.** *A priority graph $G \in \mathcal{G}$ is feasible if and only if for all elementary cycles $\mathcal{C}$ in G, we have:*

$$\bigcap_{(i,j)\in E(\mathcal{C})} \chi^{\text{obs}}_{i\succ j} = \emptyset \tag{3.12}$$

If Condition (3.12) holds, the safety margin is given by:

$$\rho_G = \max \left\{ r \geq 0 : \forall \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} \left(\chi^{\text{obs}}_{i\succ j} + [-r, r]^n \right) = \emptyset \right\} \tag{3.13}$$

A complete proof of the above theorem is provided in Appendix A.4. In the following, we prove that (3.12) is a necessary condition for priority graph feasibility, and we also provide a slightly stronger sufficient condition for priority graph feasibility.

Proof of the necessary condition. We will prove the necessary condition by contraposition. Take $G \in \mathcal{G}$ and assume that there is an elementary cycle $\mathcal{C}$ of G such that the subset $\bigcap_{(i,j)\in E(\mathcal{C})} \chi^{\text{obs}}_{i\succ j}$ is not empty, and let x^1 be an element of this set. We have

to prove that $\Phi(\chi_G^{\text{free}}) = \emptyset$. To this end, we are going to build a hyper-orthant in the coordination space capturing any path of $\Phi(\chi_G^{\text{free}})$. For each $j \in V(\mathcal{C})$, consider $K_j := \{x \in \chi : x_j = x_j^1 \text{ and } x_i \leq x_i^1 \text{ for } i \neq j\}$. By Property 2.2.1 we have:

$$x^1 \in \chi_{i \succ j}^{\text{obs}} \Longrightarrow K_j \subset \chi_{i \succ j}^{\text{obs}} \tag{3.14}$$

As $\forall (i,j) \in E(\mathcal{C}), x^1 \in \chi_{i \succ j}^{\text{obs}}$, applying the latter result yields:

$$\forall (i,j) \in E(\mathcal{C}), K_j \subset \chi_{i \succ j}^{\text{obs}} \tag{3.15}$$

As a consequence, $\bigcup_{(i,j) \in E(\mathcal{C})} K_j \subset \chi_G^{\text{obs}}$, and as $\mathcal{C}$ is a cycle, every vertex $j \in V(\mathcal{C})$ is involved in some edge $(i,j) \in E(\mathcal{C})$, so that we have:

$$\bigcup_{j \in V(\mathcal{C})} K_j \subset \chi_G^{\text{obs}} \tag{3.16}$$

In the coordination space restricted to the coordinates which appear in $\mathcal{C}$, $\bigcup_{j \in V(\mathcal{C})} K_j$ is the set of upper faces (that is the boundary) of the orthant (depicted in Figure 3.11):

$$\mathcal{O} := \{x \in \chi : \forall j \in V(\mathcal{C}), x_j \leq x_j^1\} \tag{3.17}$$

And we have by Equation (3.16):

$$\partial \mathcal{O} = \bigcup_{j \in V(\mathcal{C})} K_j \subset \chi_G^{\text{obs}} \tag{3.18}$$

Now, we will prove that $\Phi(\chi_G^{\text{free}})$ is empty by contradiction. Assume it is not empty and take an element φ of it. We must have $(\varphi(0) - \mathbb{R}_+^n) \subset \chi_G^{\text{free}}$, which implies that:

$$\varphi(0) \in \mathcal{O} \tag{3.19}$$

and $(\varphi(1) + \mathbb{R}_+^n) \subset \chi_G^{\text{free}}$, which implies:

$$\varphi(1) \notin \mathcal{O} \tag{3.20}$$

Since φ is continuous, by Lemma A.1.1 (see Appendix A.1), there exists $t \in [0,1]$ such that $\varphi(t) \in \partial \mathcal{O} = \bigcup_{j \in V(\mathcal{C})} K_j \subset \chi_G^{\text{obs}}$ and φ intersects χ_G^{obs}. It is is contradiction with $\varphi \in \Phi(\chi_G^{\text{free}})$.

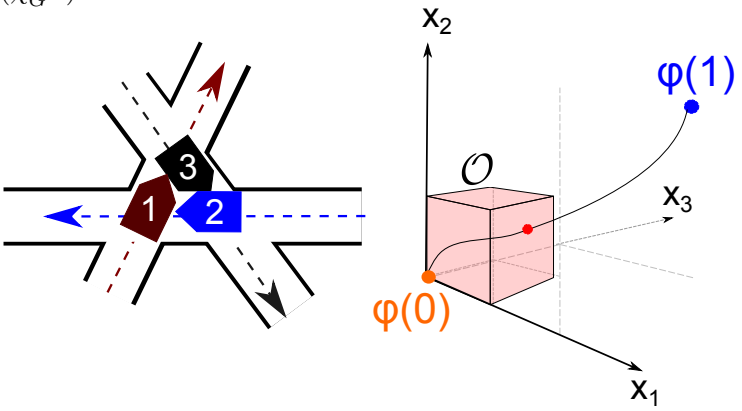


Figure 3.11: The hyper-orthant used in the proof of the necessary condition of Theorem 3.2.2 (in a three-dimensional scenario).

□

In order to provide a constructive proof of the existence of feasible paths taking values in χ_G^{free} under certain conditions, we first introduce the concept of local priority graph. Given a radius $r \geq 0$ and a configuration $x \in \chi$, the local priority graph at configuration x with radius $r \geq 0$ is the sub-graph $G_{|x,r}$ of G with the same vertices and whose edge set is defined below:

$$E(G_{|x,r}) := \left\{(i,j) \in E(G) : x \in \left(\chi_{i\succ j}^{\text{obs}} + [-r,r]^n\right)\right\} \tag{3.21}$$

As depicted in Figure 3.12, computing the local priority graph at a given configuration x with a given radius $r \geq 0$ consists in copying G and keeping only edges $(i,j) \in E(G)$ such that x belongs to the dilatation of $\chi_{i\succ j}^{\text{obs}}$ by the structuring element $[-r,r]^n$, i.e., we keep only edges (i,j) such that the distance (with the infinity norm) from x to $\chi_{i\succ j}^{\text{obs}}$ is strictly lower than r.

It is interesting to notice that in the deadlock-free example of Figure 3.12, the depicted local priority graph is acyclic. By contrast, at the deadlock configuration of Figure 3.13, even for very small radius, the local priority graph is cyclic.

Lemma 3.2.1 (Sufficient condition for locally acyclic priority graph)**.** *Consider a priority graph $G \in \mathcal{G}$ satisfying for all elementary cycles $\mathcal{C}$ in G:*

$$\bigcap_{(i,j)\in E(\mathcal{C})} \left(\chi_{i\succ j}^{\text{obs}} + [-\varepsilon,\varepsilon]^n\right) = \emptyset \tag{3.22}$$

for some $\varepsilon > 0$, then $G_{|x,\varepsilon}$ is acyclic at all configurations $x \in \chi$.

Proof. Take $G \in \mathcal{G}$ and assume Equation (3.22) is satisfied for all elementary cycles $\mathcal{C}$ in G. By construction, we have:

$$E(G_{|x,\varepsilon}) = \left\{(i,j) \in E(G) : x \in \left(\chi_{i\succ j}^{\text{obs}} + [-\varepsilon,\varepsilon]^n\right)\right\} \tag{3.23}$$

The existence of a cycle $\mathcal{C}$ in $G_{|x,\varepsilon}$ would imply that $x \in \cap_{(i,j)\in E(\mathcal{C})} \left(\chi_{i\succ j}^{\text{obs}} + [-\varepsilon,\varepsilon]^n\right)$, and would therefore contradict Equation (3.22) for this cycle. □

It is of high interest to know that the local priority graph with radius $\varepsilon > 0$ is acyclic at all configurations $x \in \chi$. Indeed, when this condition is satisfied, whatever the current configuration $x \in \chi_G^{\text{free}}$ of the system, it is always possible to find a robot $i \in \mathcal{R}$ which can move forward the distance $\varepsilon > 0$ without colliding, which enables to construct a feasible path in χ_G^{free} by iterations. Based on this idea, we propose now to give a slightly stronger sufficient condition for the existence of feasible paths satisfying a given priority graph $G \in \mathcal{G}$. We prove in the sequel that a sufficient condition for $\Phi(\chi_G^{\text{obs}}) \neq \emptyset$ is that for all elementary cycles $\mathcal{C}$ in G:

$$\bigcap_{(i,j)\in E(\mathcal{C})} \left(\chi_{i\succ j}^{\text{obs}} + [-\varepsilon,\varepsilon]^n\right) = \emptyset \tag{3.24}$$

for some $\varepsilon > 0$. It is a slightly stronger condition than in Theorem 3.2.2 as $\varepsilon > 0$ (instead of $\varepsilon \equiv 0$).

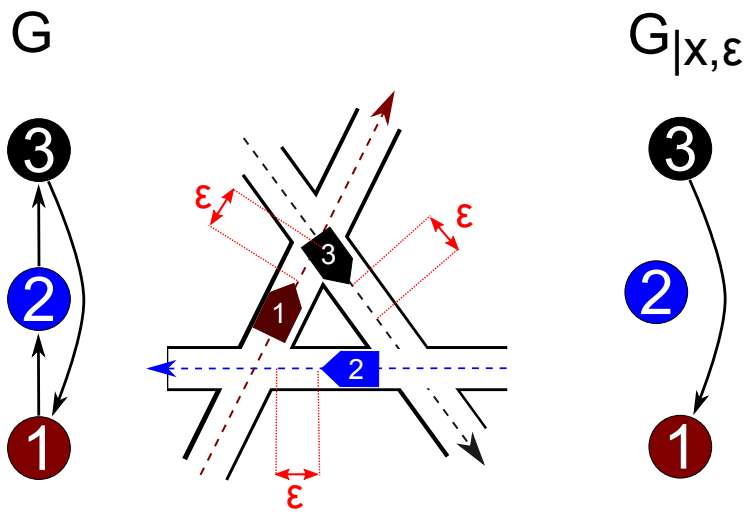


Figure 3.12: Computation of the local priority graph for a three-robot system. Note that due to the geometry of paths (in particular their relative position), for small enough radius $\varepsilon > 0$, the local priority graph is acycle at all configurations.

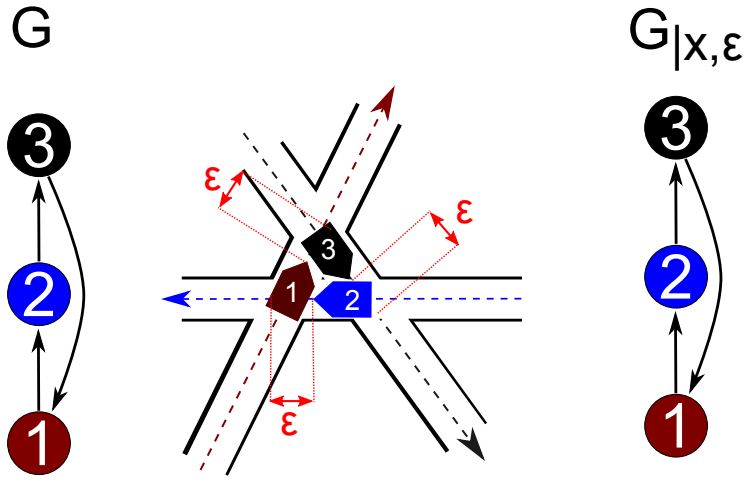


Figure 3.13: Computation of the local priority graph for a three-robot system in a deadlock configuration. Note that the local priority graph is cyclic at the deadlock configuration, even for very small radius.

Proof of the sufficient condition under a slightly stronger assumption. Take $G \in \mathcal{G}$ and assume that we have $\varepsilon > 0$ such that for all elementary cycles $\mathcal{C}$ in G, Equation (3.24) holds. We will provide a constructive proof of the existence of a path $\varphi \in \Phi(\chi_G^{\text{free}})$. By Lemma 3.2.1, the local priority graph $G_{|x,\varepsilon}$ is acyclic at all configurations $x \in \chi$. Let $x^{\text{goal}} \in \chi^{\text{free}}$ denote the desired final configuration defined componentwise as: $x_i^{\text{goal}} = \overline{x}_i^{\text{obs}} + \varepsilon$ (satisfying $\left(x^{\text{goal}} + \mathbb{R}_+^n\right) \subset \chi_G^{\text{free}}$). We define the finite time flow $\phi(x,t)$ starting at initial condition x component-wise as follows for $t \in [0,1]$ and $j \in \mathcal{R}$:

$$\phi_j(t,x) := \begin{cases} x_j & \text{if } \exists (i,j) \in E(G_{|x,\varepsilon}) \\ \min(x_j^{\text{goal}}, x_j + t\varepsilon) & \text{else.} \end{cases} \tag{3.25}$$

First we prove that the flow starting from an initial configuration in χ_G^{free} remains in χ_G^{free}. Consider $x \in \chi_G^{\text{free}}$ and $(i,j) \in E(G)$. By construction of ϕ, we have for all $t \in [0,1]$:

$$\phi_i(t,x) \geq x_i \tag{3.26}$$

For j, consider the two following options:

- $(i,j) \in E(G_{|x,\varepsilon})$. Then, we have for all $t \in [0,1]$:

  $$\phi_j(t,x) = x_j \tag{3.27}$$

  By Property 2.2.1, since Equations (3.26) and (3.27) hold, $x \in \chi_{i \succ j}^{\text{free}}$ implies that $\phi(t,x) \in \chi_{i \succ j}^{\text{free}}$.

- $(i,j) \notin E(G_{|x,\varepsilon})$. Then, we have for all $t \in [0,1]$:

  $$\phi_j(t,x) = \min(x_j^{\text{goal}}, x_j + t\varepsilon) \leq x_j + \varepsilon \tag{3.28}$$

  Moreover, by construction of the local priority graph, $(i,j) \notin E(G_{|x,\varepsilon})$ is equivalent to:

  $$x \notin \chi_{i \succ j}^{\text{obs}} + [-\varepsilon, \varepsilon]^n \tag{3.29}$$

  which implies that:

  $$x + \varepsilon \mathbf{e}_j \in \chi_{i \succ j}^{\text{free}} \tag{3.30}$$

  By Property 2.2.1, since Equations (3.26) and (3.28) hold, Equation (3.30) implies that $\phi(t,x) \in \chi_{i \succ j}^{\text{free}}$.

In conclusion, for all $x \in \chi_G^{\text{free}}$ and $t \in [0,1]$, $\phi(t,x) \in \chi_G^{\text{free}}$. Now, consider the path $\varphi(t)$ defined iteratively as follows:

$$\varphi(0) := \underline{x}^{\text{obs}} \tag{3.31}$$

$$\forall p \in \mathbb{N}, \forall t \in [0,1], \varphi(p+t) := \phi(t, \varphi(p)) \tag{3.32}$$

$\varphi(0) \in \chi_G^{\text{free}}$ and by induction, φ takes values in χ_G^{free}. It is non-decreasing as $\phi_j(t,x) \geq x_j$, and we are going to prove that it reaches x^{goal} in finite time. The

local priority graph at configuration x only contains edges (i,j) such that $x_j < x_j^{\text{goal}}$ ($x_j = x_j^{\text{goal}} = \overline{x}_j^{\text{obs}} + \varepsilon$ implies that $x \notin (\chi_{i\succ j}^{\text{obs}} + [-\varepsilon,\varepsilon]^n)$). Since the local priority graph is acyclic, for all $x \in \chi$ with $x < x^{\text{goal}}$, there exists a maximal element $j \in \mathcal{R}$ satisfying $x_j < x_j^{\text{goal}}$ and $\forall i \in \mathcal{R}$, $(i,j) \notin E(G_{|x,\varepsilon})$. By construction of ϕ, it results that for all $p \in \mathbb{N}$, if $\varphi(p) \neq x^{\text{goal}}$, then there exists at least one robot j such that $\varphi_j(p+1) = \min(x_j^{\text{goal}}, x_j + \varepsilon)$, i.e., robot j travels a distance ε or reaches its goal configuration in time interval $[p, p+1]$. The distance to travel considering all robots is finite: $\sum_{i\in\mathcal{R}} x_i^{\text{goal}} - x_i^0$. As a result, x^{goal} is reached in finite time T, $\varphi(T) = x^{\text{goal}}$ and T satisfies:

$$T \leq \left\lceil \frac{\sum_{i\in\mathcal{R}} x_i^{\text{goal}} - \underline{x}_i^{\text{obs}}}{\varepsilon} \right\rceil \tag{3.33}$$

where $\lceil . \rceil$ denotes the ceiling function. Rescaling time by a factor $1/T$ yields a path $\tilde{\varphi} \in \Phi(\chi_G^{\text{free}})$. □

3.2.4 Absence of deadlocks

To this point, we have proved a necessary and sufficient condition for $\Phi(\chi_G^{\text{free}}) \neq \emptyset$, i.e., for the existence of feasible paths whose priority is graph G. In the following, we prove that provided $\Phi(\chi_G^{\text{free}}) \neq \emptyset$, there is no deadlock configuration in χ_G^{free}. This means that for all configurations $x \in \chi_G^{\text{free}}$, there exists a path $\varphi \in \Phi(\chi_G^{\text{free}})$ going through configuration x. It is a very valuable result as a direct consequence is that provided the assigned priorities are feasible, there will be no deadlock, as long as priorities are "respected", i.e., as long the configuration of the system remains in χ_G^{free}.

Theorem 3.2.3 (Absence of deadlocks)**.** *Given $G \in \mathcal{G}$ satisfying $\Phi(\chi_G^{\text{free}}) \neq \emptyset$, for all $x \in \chi_G^{\text{free}}$, there exists $\varphi \in \Phi(\chi_G^{\text{free}})$ going through x.*

Proof. Take a priority graph $G \in \mathcal{G}$, $x \in \chi_G^{\text{free}}$, assume $\Phi(\chi_G^{\text{free}}) \neq \emptyset$ and take $\varphi \in \Phi(\chi_G^{\text{free}})$. First of all, note that concatenating φ with the segment joining $\varphi(0)$ and $\min(x, \varphi(0))$ and with the segment joining $\varphi(1)$ with $\max(x, \varphi(1))$ gives a path in $\Phi(\chi_G^{\text{free}})$ starting from a configuration lower than or equal to x and ending at a configuration greater than or equal to x. Hence, assume without loss of generality that $\varphi(0) \leq x$ and $\varphi(1) \geq x$. Define $\tilde{\varphi}^1 := \max(x, \varphi)$ and $\tilde{\varphi}^2 := \min(x, \varphi)$. These paths take values in χ_G^{free} by Property 2.2.2. The concatenation of $\tilde{\varphi}^1$ and $\tilde{\varphi}^2$ gives a path $\tilde{\varphi} \in \Phi(\chi_G^{\text{free}})$ going through x. □

Conclusions

The present part proposed a novel tool in multi robot coordination: the priority graph. It enables to go one step ahead in the understanding of the structure of the solutions to the coordination problem. Previous work noticed the existence of homotopy classes of feasible paths in the coordination space [47]. Our results demonstrate that priorities uniquely encode these homotopy classes. The existence of a finite number of homotopy classes of feasible paths then merely appears as the consequence of the finiteness of possible priority graphs. Assigning priorities plans a high-level coordination strategy represented by the priority graph describing the relative order of robots through the intersection. Under assigned priorities, the path of robots in the coordination space is just required to remain in a homotopy class of feasible paths continuously deformable into each other. Respecting assigned priorities is weaker than following a particular feasible path as a large (homotopy) class of feasible paths induce the same priorities. The size of the homotopy class provides some freedom of action. Therefore, priorities appear a relevant resource to guide robots through an intersection area. A key asset of planning priorities is that well-chosen feasible priorities – in particular, acyclic priorities – completely solves the deadlock avoidance problem (see Threorem 3.2.2). Theorem 3.2.3 proves that provided feasible priorities are respected, robots will never be stuck in a deadlock configuration. It is key to solve the deadlock avoidance problem at the planning level as deadlocks are difficult to avoid in a reactive manner.

The results of the present part are quite conceptual with little care about the dynamics model and control issues. It does not specify how to use priorities to control robots. In traditional planning, the control part consists of executing the plan by tracking the planned reference trajectory. This is known as the trajectory tracking problem [63, 80, 92, 122, 141]. The reference trajectory configures the control law which tries to minimize the tracking error (e.g., using a linear-quadratic regulator [129]). In priority-based coordination, there is no reference trajectory to track, the plan is merely the priority graph. The next part of the thesis assumes that priorities are assigned and it aims at building control laws configured by the priority graph ensuring that priorities are respected and that all robots eventually go through the intersection. From the point of view of the present part, control laws proposed in the next part ensure that the resulting path described by robots in the coordination space belongs to the homotopy class encoded by the assigned priorities.

Part II

Priority preserving control

Introduction

Previous work noticed the combinatorial complexity of multi robot control (see, e.g., [25]). In [25], the expected application is a driver assistance system to avoid crashes between human driven vehicles just in time. It is thus completely right to try to find a particular schedule to avoid the crash. In this thesis, we are in a much different context and we assume that robots are in a safe state when approaching the intersection, and a lot of different schedules – more precisely, a lot of different priority graphs – are possible to safely coordinate robots. The present part assumes that feasible priorities are assigned, that the assigned priorities are compatible with the initial state of the robots, and focuses on how to use the assigned priorities to guide robots through the intersection. As priorities are assigned, there is no combinatorial problem, and so-called priority preserving control can be carried out in polynomial time. In traditional planning, the plan is a reference trajectory which configures a control law in charge of tracking the reference trajectory. In priority-based coordination, the plan is the priority graph, so the control law is configured by the priority graph and is in charge of ensuring priority preservation (no collision occurs and priorities are respected). Ensuring priority preservation is much weaker than tracking a reference trajectory, so robots retain some freedom of action. The proposed control law guarantees liveness, i.e., following the control law, all robots eventually go through the intersection. The freedom of action enabled by planning only priorities is highlighted, as under the presented control law, robots may brake at any point of time without violating priorities, in particular without colliding. This robustness property is quite novel among existing coordination systems and is highly valuable as it is very likely to happen that robots need to brake to handle some unexpected event (e.g., a pedestrian crossing the road, a loss of communication abilities, a congestion at the exit of the intersection). Finally, the proposed control scheme in Chapters 5 and 6 is decentralized as the output of the control law can be computed on each robot independently without an agreement through communication links.

Sketch of the part The assigned priorities are assumed to be acyclic. Under this assumption, Chapter 4 provides a priority preserving control law for robots controlled in velocity; Chapter 5 examines the case of robots controlled in acceleration; and in Chapter 6, robustness of priority preserving control with respect to bounded noise is illustrated. The reader is referred to Appendix B for an extension of the results of this part to feasible cyclic priority graphs under mild assumptions.

Note to the reader The two first chapters of the present part are independent. However, it is advised to start with the first chapter for a gradual understanding of the proposed method. The last chapter is not necessary to the understanding of the rest of the thesis. The reader without special interest in considering uncertainty concerns can skip Chapter 6 and go directly to Part III.

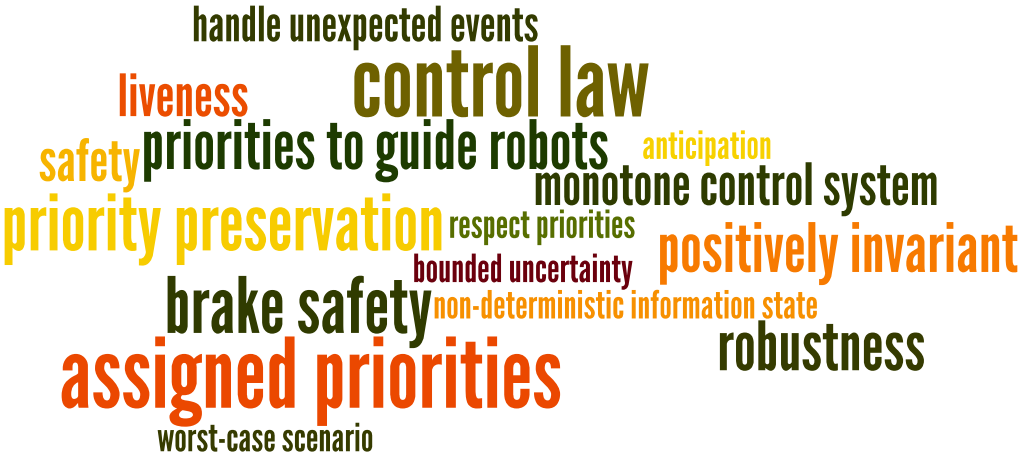

handle unexpected events
control law
liveness
priorities to guide robots
anticipation
safety
monotone control system
priority preservation
respect priorities
positively invariant
bounded uncertainty
brake safety
non-deterministic information state
robustness
assigned priorities
worst-case scenario

Table of Contents

4 Priority preserving control in the absence of inertia **63**

4.1 A monotone control system 63

4.2 The proposed control law 64

4.2.1 Priority preservation 65

4.2.2 Optimality 68

4.2.3 Liveness 70

5 Priority preserving control under kinodynamic constraints **71**

5.1 A monotone control system 71

5.2 The proposed decentralized control law 73

5.2.1 Priority preservation 76

5.2.2 Robustness 78

5.2.3 Liveness 81

6 Robustness with respect to bounded noise **83**

6.1 Control model with bounded noise 84

6.2 Evolution of the non-deterministic information state 85

6.3 The proposed decentralized control law 87

6.3.1 Priority preservation 87

6.3.2 Robustness 88

6.3.3 Liveness 90

Chapter 4

Priority preserving control in the absence of inertia

Contents

4.1 A monotone control system 63
4.2 The proposed control law 64
4.2.1 Priority preservation 65
4.2.2 Optimality 68
4.2.3 Liveness 70

In the present chapter, the velocity of the robots is assumed to be controlled, and a control law aimed at coordinating multiple robots with assigned priorities is proposed.

Sketch of the chapter Section 4.1 exposes the dynamics model and shows that the resulting system is a monotone control system [9]. Section 4.2 constructs a priority preserving control law. Optimality and liveness properties are provided.

4.1 A monotone control system

Each robot i is modeled as a first-order control system with state $x_i \in \mathbb{R}$, whose evolution is described by the differential equation:

$$\dot{x_i}(t) = \mathbf{v}_i(t) \tag{4.1}$$

where $\mathbf{v}_i : \mathbb{R}_+ \to V_i$ is the control of robot i. We let $V_i := \{0, \overline{v}_i\}$ be the set of feasible control values. The control is assumed to be updated in discrete time every $\Delta t > 0$:

$$\forall k \in \mathbb{N}, \forall t \in [k\Delta t, (k+1)\Delta t), \mathbf{v}_i(t) \equiv \mathbf{v}_i(k\Delta t) \tag{4.2}$$

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 $\mathbf{V}_i$ denote the set of controls $\mathbf{v}_i : \mathbb{R}_+ \to V_i$ piecewise constant on intervals $[k, k+1)$, $k \in \mathbb{N}$. We let $t \mapsto \phi_i(t, x_i, \mathbf{v}_i)$ denote the flow of the system starting at initial configuration $x_i \in \mathbb{R}$ with control $\mathbf{v}_i \in \mathbf{V}_i$ as depicted in Figure 4.1.

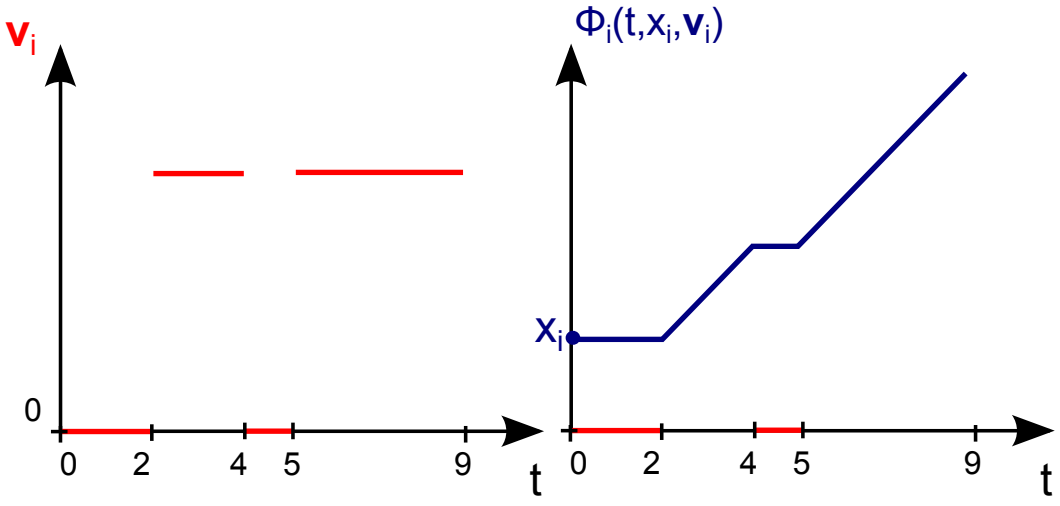


Figure 4.1: An example of piecewise constant velocity control $\mathbf{v}_i$ (left) and the corresponding flow $t \mapsto \phi_i(t, x_i, \mathbf{v}_i)$ starting from the initial configuration x_i (right).

We also define the vectorial control $\mathbf{v} := (\mathbf{v}_i)_{i\in\mathcal{R}} \in \mathbf{V} := \prod_{i\in\mathcal{R}} \mathbf{V}_i$, and the vectorial flow: $\phi(t, x, \mathbf{v}) := (\phi_i(t, x_i, \mathbf{v}_i))_{i\in\mathcal{R}}$. We let $\overline{v} := (\overline{v}_i)_{i\in\mathcal{R}}$ and we define the constant control $\overline{\mathbf{v}}(t) := \overline{v}$. We introduce partial orders as follows:

$$\forall \mathbf{v}_i^1, \mathbf{v}_i^2 \in \mathbf{V}_i, \mathbf{v}_i^1 \preceq \mathbf{v}_i^2 \quad \text{if} \quad \forall t \geq 0, \mathbf{v}_i^1(t) \leq \mathbf{v}_i^2(t) \tag{4.3}$$

$$\forall \phi^1, \phi^2 : \mathbb{R}_+ \to \chi, \phi^1 \preceq \phi^2 \quad \text{if} \quad \forall t \geq 0, \phi^1(t) \preceq \phi^2(t) \tag{4.4}$$

The control system (4.1) is a monotone control system [9] with regards to the relative orders defined above. More precisely, the following key property holds:

Property 4.1.1 (Order preservation)**.** *The flow $t \mapsto \phi_i(t, x_i, \mathbf{v}_i)$ is order-preserving with regards to x_i and $\mathbf{v}_i$.*

Note that in our open loop model, control $\mathbf{v}_i$ only acts on robot i, that is, $\mathbf{v}$ 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.

4.2 The proposed control law

Now, we propose to build a control law $f^G : \chi \to V$ such that starting from an initial collision-free configuration, the flow of the system controlled by the control law f^G is ensured to remain in χ_G^{free} (thus being collision-free and respecting priorities G). In other words, using the terminology of [68], χ_G^{free} shall be positively invariant for the system under control law f^G.

The rationale for our control law is as follows. Each robot $i \in \mathcal{R}$ moves forward, unless moving forward violates the priority with regards to some robot $j \in \mathcal{R}$ with

$(j,i) \in E(G)$. In the coordination space, violating such a priority means that the configuration of the system would collide with $\chi^{\text{obs}}_{j \succ i}$. The control law can then be formulated synthetically component-wise:

$$f^G_i(x) := \begin{cases} 0 & \text{if } \exists (j,i) \in E(G), \exists t \in [0,1] \text{ s.t. } \left(x + t\left(\overline{v}_i \mathbf{e}_i + f^G_j(x)\mathbf{e}_j\right)\right) \in \chi^{\text{obs}}_{j \succ i} \\ \overline{v}_i & \text{else.} \end{cases} \tag{4.5}$$

First of all, note that f^G appears in both the left-hand side and the right-hand side in Equation (4.5). Hence, it is not obvious that Equation (4.5) effectively defines a control law which is stated by the following theorem. Note that a decentralized version of the proposed control law could be used alternatively by considering the worst case scenario for each robots $(j,i) \in E(G)$, i.e., when robot j stops (see the decentralized control law of Chapter 5). However, the optimality result that we obtain in the present chapter would not hold anymore.

Theorem 4.2.1 (Control law existence)**.** *Given an acyclic priority graph G, Equation (4.5) uniquely defines a control law $f^G : \chi \to V$.*

Proof. The priority graph is assumed to be acyclic. Hence, there exists a topological ordering of the graph such that for every edge $(j,i) \in E(G)$, j comes before i in the ordering. Following the topological order induced by G, it is possible to compute $f^G_i(x)$ for all $i \in \mathcal{R}$ iteratively. As a result, Equation (4.5) uniquely defines a control law $f^G : \chi \to V$. □

Figure 4.2 and 4.3 show the evolution of a three-robot system under control law f^G under acyclic priorities. It is clear in Figure 4.3 that the control law belongs to the "bug" family, emanating from the work of [87]. Indeed, the robots go at maximum speed until they are too close to the boundary of the obstacle region. Then, they follow the boundary with a certain distance as long as necessary.

Now, we need to introduce the following notation. Given a feedback control law $f : \chi \to V$, with a slight abuse of notation we let $t \mapsto \phi(t, x, f)$ denote the vectorial flow of the system starting at initial condition $x \in \chi$ and controlled by $\mathbf{v} \in \mathbf{V}$ satisfying:

$$\forall k \in \mathbb{N}, \mathbf{v}(k) \equiv f(\phi(k, x, \mathbf{v})) \tag{4.6}$$

4.2.1 Priority preservation

First of all, we prove the key property of our control law that is the safety guarantee. More precisely, starting from a configuration in χ^{free}_G, the system under control law f^G is ensured to remain in χ^{free}_G, i.e., priorities G are preserved. Following the terminology of [68], χ^{free}_G is positively invariant under control law f^G as stated in the following theorem:

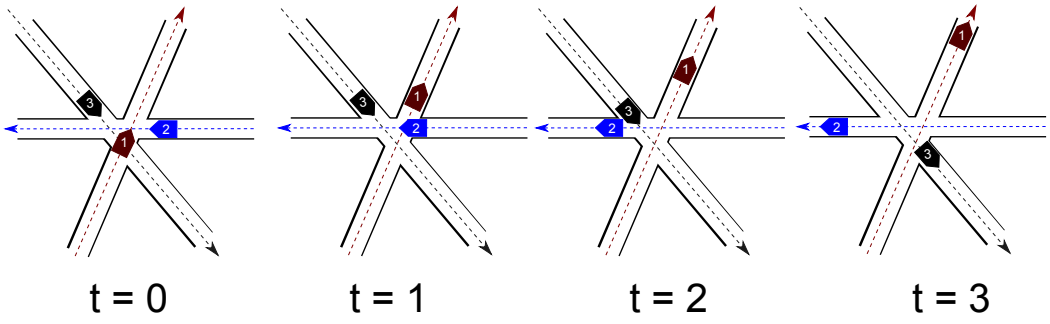


Figure 4.2: A three-robot system with acyclic assigned priorities: $1 \succ 2$, $2 \succ 3$, *and* $1 \succ 3$. *Robots are controlled under control law* f^G. *The drawings show the evolution of the robots along their paths.*

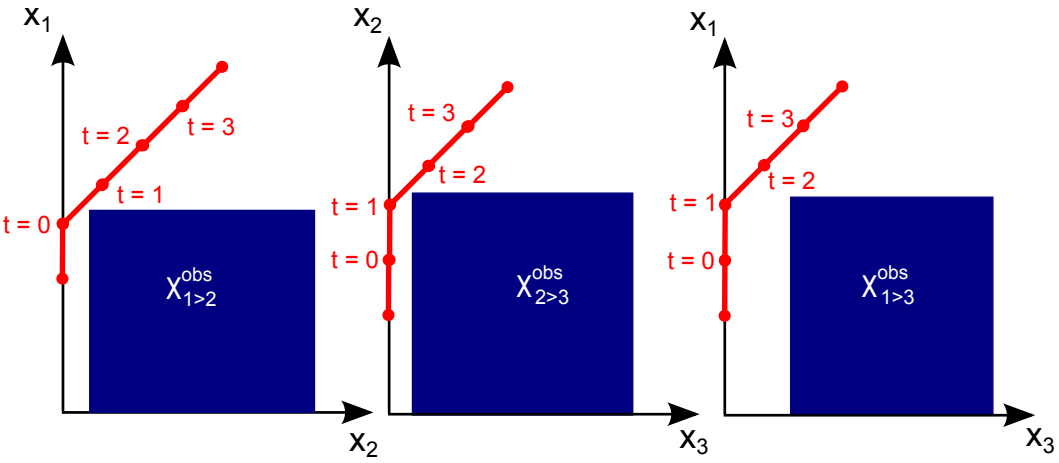


Figure 4.3: Trajectory in the coordination space for the three-robot system under control law f^G *of Figure 4.2.*

Theorem 4.2.2 (Priority preservation)**.** *Given an acylic priority graph G, χ_G^{free} is positively invariant for the system under control law f^G, i.e.,*

$$\forall x \in \chi_G^{\text{free}}, \forall t \geq 0, \phi(t, x, f^G) \in \chi_G^{\text{free}} \tag{4.7}$$

Proof. Take an acyclic priority graph G and an initial configuration $x \in \chi_G^{\text{free}}$. By induction, it is sufficient to prove that the flow satisfies:

$$\forall t \in [0,1], \phi(t, x, f^G) \in \chi_G^{\text{free}} \tag{4.8}$$

By construction, for all $i \in \mathcal{R}$ and for all $(j,i) \in E(G)$, we have two options:

- either $f_i^G(x) = 0$. For all $t \in [0,1]$, we have:

$$\begin{aligned} x_i + t f_i^G(x) &= x_i & (4.9)\\ x_j + t f_j^G(x) &\geq x_j & (4.10) \end{aligned}$$

  Hence, by Property 2.2.1, $x \in \chi_{j \succ i}^{\text{free}}$ implies that for all $t \in [0,1]$, $\left(x + t\left(f_i^G(x)\mathbf{e}_i + f_j^G(x)\mathbf{e}_j\right)\right) \in \chi_{j\succ i}^{\text{free}}$.

- or $f_i^G(x) = \overline{v}_i$. Then, by construction of the control law, we have for all $t \in [0,1]$, $\left(x + t\left(f_i^G(x)\mathbf{e}_i + f_j(x)^G\mathbf{e}_j\right)\right) \equiv \left(x + t\left(\overline{v}_i\mathbf{e}_i + f_j^G(x)\mathbf{e}_j\right)\right) \in \chi_{j\succ i}^{\text{free}}$.

Hence, in both cases, we obtain:

$$x + t\left(f_i^G(x)\mathbf{e}_i + f_j^G(x)\mathbf{e}_j\right) \in \chi_{j\succ i}^{\text{free}} \tag{4.11}$$

Moreover, we have:

$$\begin{aligned} \phi_i(t, x, f^G) &= x_i + t f_i^G(x) & (4.12)\\ \phi_j(t, x, f^G) &= x_j + t f_j^G(x) & (4.13) \end{aligned}$$

As a result, Equation (4.11) implies that $\phi(t, x, f^G) \in \chi_{j\succ i}^{\text{free}}$ for all $(j,i) \in E(G)$ and $t \in [0,1]$, i.e., $\phi(t, x, f^G) \in \chi_G^{\text{free}}$ for all $t \in [0,1]$. □

Given a configuration $x \in \chi_G^{\text{free}}$ and a priority graph G, we say $\mathbf{v}$ is a collision-free control for the pair (x, G) if the flow starting from $x \in \chi_G^{\text{free}}$ remains in χ_G^{free}. We write $\mathbf{v} \in \mathbf{V}_G^{\text{free}}(x)$ defined as follows:

$$\mathbf{V}_G^{\text{free}}(x) := \left\{\mathbf{v} \in \mathbf{V} : \phi(\mathbb{R}_+, x, \mathbf{v}) \subset \chi_G^{\text{free}}\right\} \tag{4.14}$$

4.2.2 Optimality

First of all, we define the notion of optimality under assigned priorities used in the sequel. Given a priority graph G and a control law f, we say f is optimal for the priority graph G if for all configurations $x \in \chi_G^{\text{free}}$ and for all controls $\mathbf{v} \in \mathbf{V}_G^{\text{free}}(x)$, we have:

$$\forall t \geq 0, \phi(t, x, f) \geq \phi(t, x, \mathbf{v}) \tag{4.15}$$

In other words, the control law is optimal if for each robot, it maximizes the distance travelled through time while respecting priorities G. Note that this kind of optimality is even stronger than the family of Pareto optimality. Pareto optimality would state that it is impossible to make any individual robot travel farther without making at least one robot travel less. By contrast, our optimality result states that even if other robots travel less, it's impossible to make one robot travel farther while respecting the assigned priorities, i.e., all individual objectives are optimized. As a consequence, the obtained trajectory is optimal for a whole set of utility functions, more precisely, all utility functions which grow with the distance traveled by robots. For example, it minimizes the average exit time of robots, it also minimizes the maximum exit time of robots (the time at which the last robot exits the intersection). However, it is important to note that the optimality result is conditioned on the assigned priorities. Note that the trajectory resulting from the application of the proposed control law corresponds to the left-greedy optimal trajectory of References [47, 46], where it is noticed that it is a local optimum, in that it is optimal over trajectories belonging to the same homotopy class. Hence, obtaining a globally optimal trajectory would require exploring all feasible priorities, i.e., exploring all homotopy classes.

Theorem 4.2.3 (Optimality)**.** *Given an acyclic priority graph G, the control law f^G is optimal for the priority graph G, in the sense that for all controls $\mathbf{v} \in \mathbf{V}_G^{\text{free}}(x)$, we have:*

$$\forall t \geq 0, \phi(t, x, f^G) \geq \phi(t, x, \mathbf{v}) \tag{4.16}$$

Proof. We will prove Theorem 4.2.3 by contraposition. Take an acyclic priority graph G, an initial condition $x \in \chi^{\text{free}}$ and a control $\mathbf{v} \in \mathbf{V}$, Assume that there exists $i \in \mathcal{R}$ and $t \geq 0$ such that $\phi_i(t, x, f^G) < \phi_i(t, x, \mathbf{v})$. We have to prove that $\phi(\mathbb{R}_+, x, \mathbf{v}) \cap \chi_G^{\text{obs}} \neq \emptyset$. Consider $I := \{t \geq 0 : \exists i \in \mathcal{R} : \phi_i(t, x, f^G) < \phi_i(t, x, \mathbf{v})\}$. By assumption, $I \neq \emptyset$, then I is a lower-bounded non-empty subset of $\mathbb{R}$, so that $t^0 = \inf I$ exists. Let k^0 be the unique $k \in \mathbb{N}$ such that $t^0 \in [k, k+1)$. By definition of t^0 and as the velocity control is piece-wise constant, we have:

$$\forall j \in \mathcal{R}, \phi_j(k^0, x, f^G) \geq \phi_j(k^0, x, \mathbf{v}) \tag{4.17}$$

and there exists $i \in \mathcal{R}$ such that:

$$\phi_i(k^0 + 1, x, f^G) < \phi_i(k^0 + 1, x, \mathbf{v}) \tag{4.18}$$

As $V_i = \{0, \overline{v}_i\}$ (binary velocity control), Equations (4.18) and (4.17) imply that:

$$\phi_i(k^0, x, f^G) = \phi_i(k^0, x, \mathbf{v}) \tag{4.19}$$

$$\mathbf{v}_i(k) = \overline{v}_i > f_i^G(x^0) = 0 \tag{4.20}$$

where $x^0 := \phi(k^0, x, f^G)$. As $f_i^G(x^0) = 0$, by construction of the control law f^G, there is necessarily an edge (j, i) in the graph G satisfying:

$$\left(x^0 + t\left(\overline{v}_i \mathbf{e}_i + f_j^G(x^0)\mathbf{e}_j\right)\right) \in \chi^{\text{obs}}_{j \succ i} \tag{4.21}$$

for some $t \in [0, 1]$. Assume additionally that i is chosen to be a maximal element of the (acyclic) sub-graph of G containing only vertices satisfying Equation (4.18). Then, as $(j, i) \in E(G)$, j does not satisfy Equation (4.18) and we have:

$$\phi_j(k^0 + t, x, f^G) \geq \phi_j(k^0 + t, x, \mathbf{v}) \tag{4.22}$$

Combining Equations (4.19) and (4.22), we obtain:

$$x_i^0 + t\overline{v}_i \quad = \phi_i(k^0, x, \mathbf{v}) + t\overline{v}_i = \quad \phi_i(k^0 + t, x, \mathbf{v}) \tag{4.23}$$

$$x_j^0 + t f_j^G(x^0) \quad = \phi_j(k^0 + t, x, f^G) \geq \quad \phi_j(k^0 + t, x, \mathbf{v}) \tag{4.24}$$

By Property 2.2.1, as Equations (4.23) and (4.24) are satisfied, Equation (4.21) implies that:

$$\phi(k^0 + t, x, \mathbf{v}) \in \chi^{\text{obs}}_{j \succ i} \subset \chi^{\text{obs}}_G \tag{4.25}$$

In conclusion,

$$\phi(\mathbb{R}_+, x, \mathbf{v}) \cap \chi^{\text{obs}}_G \neq \emptyset \tag{4.26}$$

□

The above theorem is illustrated in Figure 4.4. It is clear that if at some point

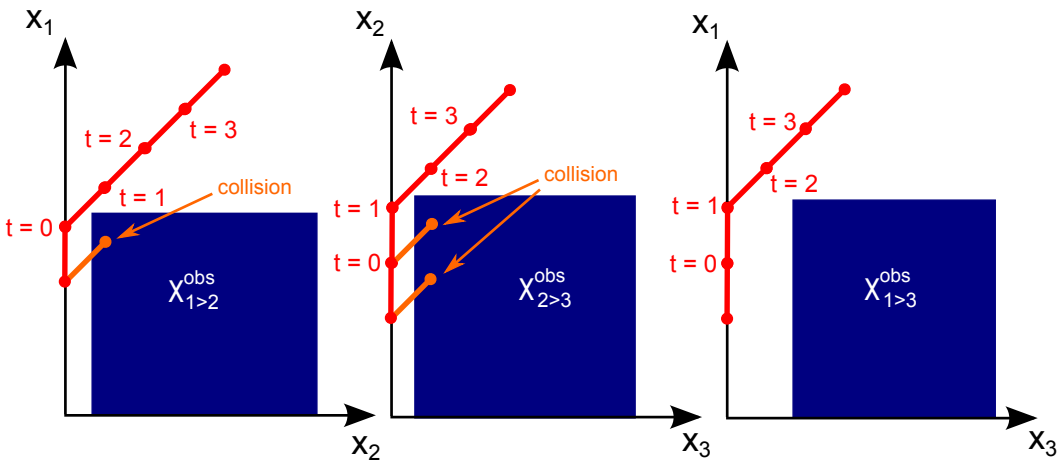


Figure 4.4: Illustration of the optimality of the control law for a three-robot system.

of time one robot tries to go faster than prescribed by the control law, a collision would occur. For example, in the left drawing, one can see that if robot 3 tries to move forward at time $t = 0$ instead of stopping as prescribed by the control law, a collision occurs (see the orange segment).

4.2.3 Liveness

A 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. Hence, liveness is guaranteed if every robot $i \in \mathcal{R}$ eventually reaches the region $\chi^{\text{goal}} := \overline{x}^{\text{obs}} + \mathbb{R}^n_+$.

Theorem 4.2.4 (Liveness)**.** *Given an acyclic priority graph G and a configuration $x^0 \in \chi^{\text{free}}_G$, there exists $T > 0$ such that:*

$$\phi(T, x^0, f^G) \in \chi^{\text{goal}} \tag{4.27}$$

The idea of the proof is that under acyclic priorities there is always at least one non exited robot able to move forward at maximum velocity until it exits the intersection.

Proof. Take an acyclic priority graph G. Consider the trajectory of the robots under control law f^G. G being acyclic, there exists an extremal vertex $i_1 \in \mathcal{R}$ such that for all $j \in \mathcal{R}$, $(j, i_1) \notin E(G)$. As a result, under the control law f^G, robot i_1 will always travel at maximal velocity and it will exit the intersection (it will reach position $\overline{x}^{\text{obs}}_i$) 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 $i_{m+1} \in \mathcal{R} \setminus \{i_1 \cdots i_m\}$ such that for all $j \in \mathcal{R} \setminus \{i_1 \cdots i_m\}$, $(j, i_{m+1}) \notin E(G)$. Collisions occurring only with non exited robots, for $t \geq T_m$ j will always be at maximum velocity and it will exit the intersection in finite time at instant $T_{m+1} \geq T_m$.

Iterating this process yields a sequence $(T_1 \cdots T_n)$ and all robots have exited the intersection at time $T := T_n$. □

Chapter 5

Priority preserving control under kinodynamic constraints

Contents

5.1 A monotone control system 71
5.2 The proposed decentralized control law 73
5.2.1 Priority preservation 76
5.2.2 Robustness 78
5.2.3 Liveness 81

In the present chapter, the acceleration of the robots is assumed to be controlled, and a control law is proposed aiming at coordinating multiple robots with assigned priorities under second-order kinodynamic constraints. The method is inspired by References [28, 25, 71, 136, 55] dealing with the coordination of a small number of vehicles without explicit notion of priorities. Preliminaries of the presented results are presented in our conference paper [50], and a more accomplished version in our article [49].

Sketch of the chapter Similarly to the previous chapter, the first section exposes the second-order dynamics model and shows that the resulting system is a monotone control system. The second section constructs a priority preserving control law. Liveness is provably guaranteed and a robustness property is provided stating that a robot may brake at any moment with neither colliding, nor violating priorities.

5.1 A monotone control system

Each robot i is modeled as a second-order control 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:

$$\dot{x}_i(t) = v_i(t) \tag{5.1}$$
$$\dot{v}_i(t) = \mathbf{u}_i(t)\ \delta(\mathbf{u}_i(t), v_i(t)) \tag{5.2}$$

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 $\Delta t > 0$:

$$\forall k \in \mathbb{N}, \forall t \in [k\Delta t, (k+1)\Delta t), \mathbf{u}_i(t) \equiv \mathbf{u}_i(k\Delta t) \tag{5.3}$$

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 $s := (s_i)_{i\in\mathcal{R}} \in S$, the vectorial control $\mathbf{u} := (\mathbf{u}_i)_{i\in\mathcal{R}} \in \mathbf{U} := \prod_{i\in\mathcal{R}} \mathbf{U}_i$, and the vectorial flow: $\phi(t, s, \mathbf{u}) := (\phi_i(t, s_i, \mathbf{u}_i))_{i\in\mathcal{R}}$. We let $\underline{u} := (\underline{u}_i)_{i\in\mathcal{R}}$, $\overline{u} := (\overline{u}_i)_{i\in\mathcal{R}}$ and we define the constant controls $\underline{\mathbf{u}}(t) := \underline{u}$ and $\overline{\mathbf{u}}(t) := \overline{u}$.

We define projection operators as follows: given a state $s = (x, v) = (s_i)_{i\in\mathcal{R}} = ((x_i, v_i))_{i\in\mathcal{R}}$, we let $\pi_x(s) := x$, $\pi_{x,i}(s) := \pi_{x,i}(s_i) := x_i$, $\pi_v(s) := v$, and $\pi_{v,i}(s) := \pi_{v,i}(s_i) := v_i$. We also define projected flows as follows: $\phi_x = \pi_x \circ \phi$, $\phi_{x,i} = \pi_{x,i} \circ \phi$, $\phi_v = \pi_v \circ \phi$ and $\phi_{v,i} = \pi_{v,i} \circ \phi$. Figure 5.1 depicts the projected flow $t \mapsto \phi_{x,i}(t, s_i, \mathbf{u}_i)$ for a particular control $\mathbf{u}_i$.

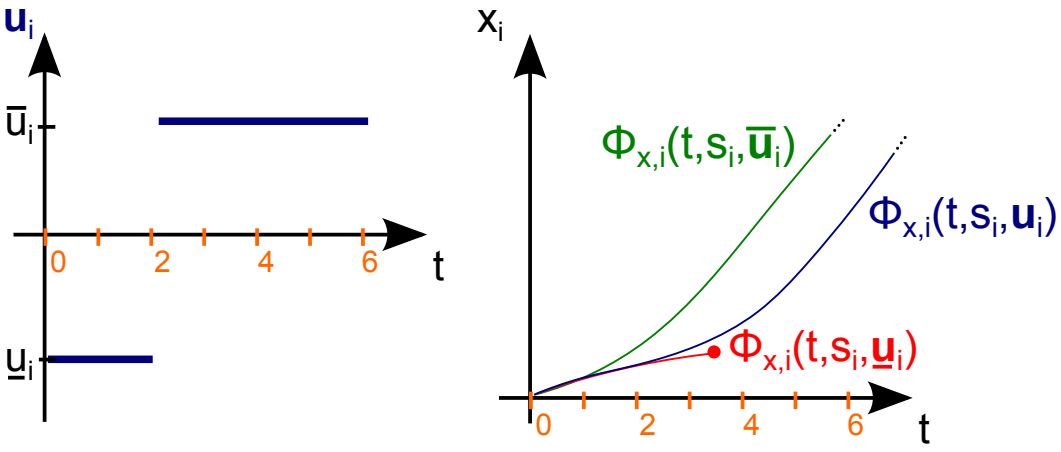


Figure 5.1: An example of piecewise constant control $\mathbf{u}_i$ (left) and the corresponding projected flow $t \mapsto \phi_{x,i}(t, s_i, \mathbf{u}_i)$ starting from the initial configuration s_i (right).

We introduce partial orders as follows:

$$\forall \mathbf{u}_i^1, \mathbf{u}_i^2 \in \mathbf{U}_i, \mathbf{u}_i^1 \preceq \mathbf{u}_i^2 \quad \text{if} \quad \forall t \geq 0, \mathbf{u}_i^1(t) \leq \mathbf{u}_i^2(t) \tag{5.4}$$

$$\forall s_i^1 = (x_i^1, v_i^1), s_i^2 = (x_i^2, v_i^2) \in S_i, s_i^1 \preceq s_i^2 \quad \text{if} \quad x_i^1 \leq x_i^2 \text{ and } v_i^1 \leq v_i^2 \tag{5.5}$$

$$\forall \phi^1, \phi^2 : \mathbb{R}_+ \to S, \phi^1 \preceq \phi^2 \quad \text{if} \quad \forall t \geq 0, \phi^1(t) \preceq \phi^2(t) \tag{5.6}$$

The control system (5.1)-(5.2) is a monotone control system [9] with regards to the relative orders defined above as easily seen in Figure 5.1 (in this example, we have $\underline{\mathbf{u}}_i \preceq \mathbf{u}_i \preceq \overline{\mathbf{u}}_i$). More precisely, the following key property holds:

Property 5.1.1 (Order preservation)**.** *The flow $t \mapsto \phi_i(t, s_i, \mathbf{u}_i)$ is order-preserving with regards to s_i and $\mathbf{u}_i$.*

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.

5.2 The proposed decentralized control law

In the absence of inertia as in Chapter 4, robots can stop instantly to respect priorities. With second-order dynamics, robots cannot stop instantly anymore and need to anticipate, taking into account their brake distance, to effectively respect priorities. The idea proposed here is to constrain the multi robot system to remain in so-called brake safe states where robots can always safely brake without colliding.

Define the set of brake safe states as follows:

$$B_G := \{s \in S : \phi_x(\mathbb{R}_+, s, \underline{\mathbf{u}}) \subset \chi_G^{\text{free}}\} \subset S \tag{5.7}$$

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} (see Figure 5.2). 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. Figure 5.2 illustrates brake safety in the coordination space and Figure 5.3 attempts to represent the concept in the real space. Brake safety is more conservative than remaining in the escape set proposed in [28], 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 [41, 20] where neither the geometric path in $\mathbb{R}_2$ nor the control to avoid collisions are fixed. The idea behind this quite conservative approach is twofold:

(a) designing a decentralized control law: the output of the control law proposed in the following can be computed independently on each robot. It is much different from approaches where collision-free trajectories are computed, either in a centralized manner (see, e.g., [25] where decentralization is considered as a possible extension), or with some agreement with message-passing through communication links (see, e.g., [13]).

(b) demonstrating robustness regarding unexpected deceleration of some robots: we believe that this is a highly valuable property as many unpredictable events requiring a robot to brake may happen in real applications.

Importantly, note that checking whether a state $s \in S$ is brake safe consists in computing a finite time single flow $t \mapsto \phi(t, s, \underline{u})$ and checking for collisions with

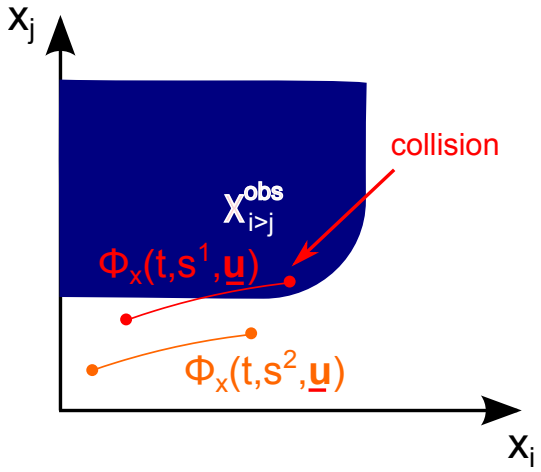


Figure 5.2: Illustration of brake safety in the coordination space for a two-robot scenario with the assigned priority $(i,j) \in E(G)$. The flow starting from s^1 (resp. s^2) under control $\underline{\mathbf{u}}$ is constructed. s^1 is not brake safe as this flow collides $\chi^{\text{obs}}_{i \succ j}$. s^2 is brake safe as the flow is collision-free with regards to $\chi^{\text{obs}}_{i \succ j}$.

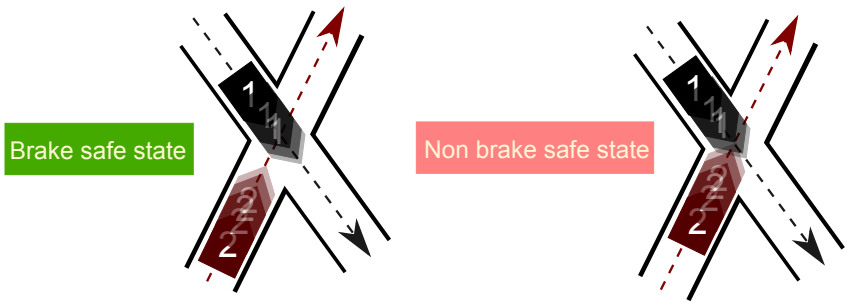


Figure 5.3: Illustration of brake safety in the real space for a two-robot scenario. Robots with lower opacity are used to represent the flow under maximum brake command. In the left drawing, the two robots stop without colliding when applying maximum brake command: they are in a brake safe state. In the right drawing, a collision occurs when the two robots brake maximally: they are not in a brake safe state.

respect to each completed obstacle region $\chi^{\text{obs}}_{i \succ j}$ for all $(i,j) \in E(G)$, yielding a quadratric complexity.

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 [68], B_G shall be positively invariant for the system under control law g^G.

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 this 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 (see Figure 5.4):

$$\mathbf{u}_i^{\text{impulse}}(k) := \begin{cases} \overline{u}_i & \text{if } k = 0 \\ \underline{u}_i & \text{if } k \geq 1 \end{cases} \tag{5.8}$$

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

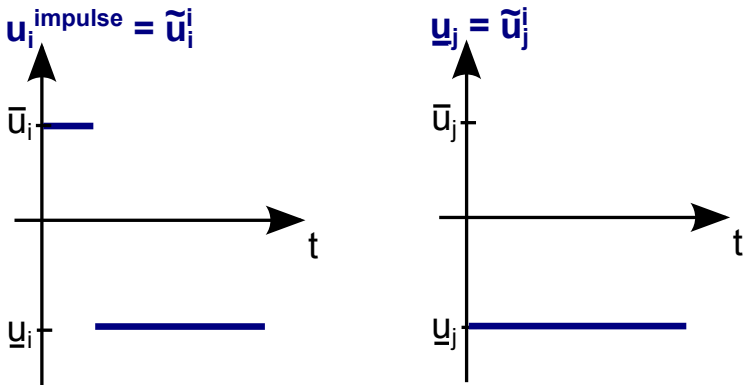


Figure 5.4: The control $\tilde{\mathbf{u}}^i$ used in the formulation of the control law. For $j \neq$ ß (right drawing), $\tilde{\mathbf{u}}^i_j$ is simply the maximum brake command $\underline{\mathbf{u}}_j$. For $j = i$ (left drawing), $\tilde{\mathbf{u}}^i_i$ is the impulse control $\mathbf{u}_i^{\text{impulse}}$.

ponentwise by (see Figure 5.4):

$$\tilde{\mathbf{u}}^i_j := \begin{cases} \mathbf{u}_i^{\text{impulse}} & \text{if } j = i \\ \underline{\mathbf{u}}_j & \text{if } j \neq i \end{cases} \tag{5.9}$$

The control law can then be formulated synthetically:

$$g_i^G(s) := \begin{cases} \underline{u}_i & \text{if } \exists (j,i) \in E(G), \exists t \geq 0 \text{ s.t. } \phi_x(t, s, \tilde{\mathbf{u}}^i) \in \chi^{\text{obs}}_{j \succ i} \\ \overline{u}_i & \text{else.} \end{cases} \tag{5.10}$$

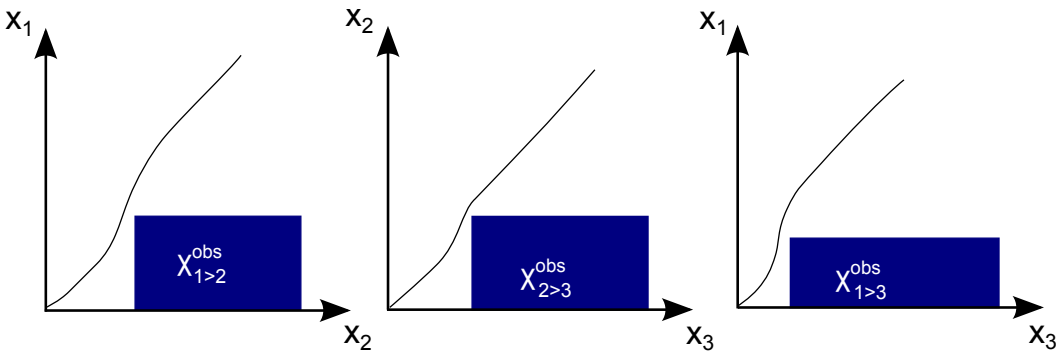


Figure 5.5: Look of the trajectory for a three-robot system with acyclic assigned priorities $1 \succ 2$, $2 \succ 3$ *and* $1 \succ 3$ *under control law* g^G.

This simply means that robot i always keeps a safe distance so that if a higher-priority robot j suddenly brakes, robot i may apply the maximum brake command until possibly stop without violating the priority. To this purpose, robot i looks at the state that would be reached if it accelerates while the higher-priority robot j brakes. If the simulated reached state is brake safe, i may accelerate; otherwise, it must brake (see the two cases in Figures 5.6 and 5.6). The look of the trajectory under control law g^G is depicted in Figure 5.5. Note that, as for brake safety checking, computing the output of the control law is of quadratic complexity for the same reasons: it requires to compute a finite time single flow and to check for collisions. Note also that each component $g_i^G(s)$ can be computed independently for each $i \in \mathcal{R}$, which means that the proposed control law is decentralized.

5.2.1 Priority preservation

Now, we need to introduce the following notation. Given a feedback control law $g : S \to U$, with a slight abuse of notation we let $t \mapsto \phi(t, s, g)$ denote the vectorial flow of the system starting at initial condition $s \in S$ and controlled by $\mathbf{u} \in \mathbf{U}$ satisfying:

$$\forall k \in \mathbb{N}, \mathbf{u}(k) \equiv g(\phi(k, s, \mathbf{u})) \tag{5.11}$$

First of all, we prove the key property of our control law that is the safety guarantee. More precisely, starting from a brake safe state in B_G, the system under control law g^G is ensured to remain in B_G, i.e., priorities G are preserved and the system is always in a brake safe state. Following the terminology of [68], B_G is positively invariant under control law g^G as stated in the following theorem:

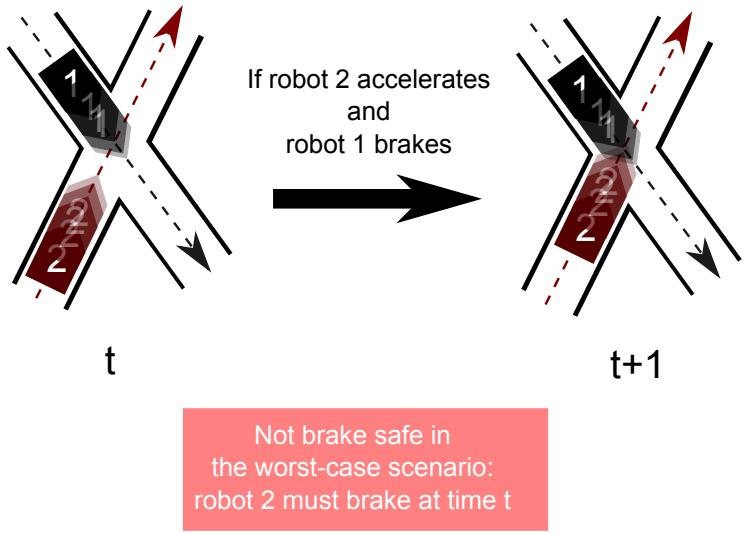


Figure 5.6: In this setting where robot 1 is assumed to have priority, the worst-case scenario under which robot 2 accelerates and robot 1 brakes leads to a new state which is not brake safe. In this case, the control law requires robot 2 to brake. Robots with lower opacity are used to represent the brake trajectory.

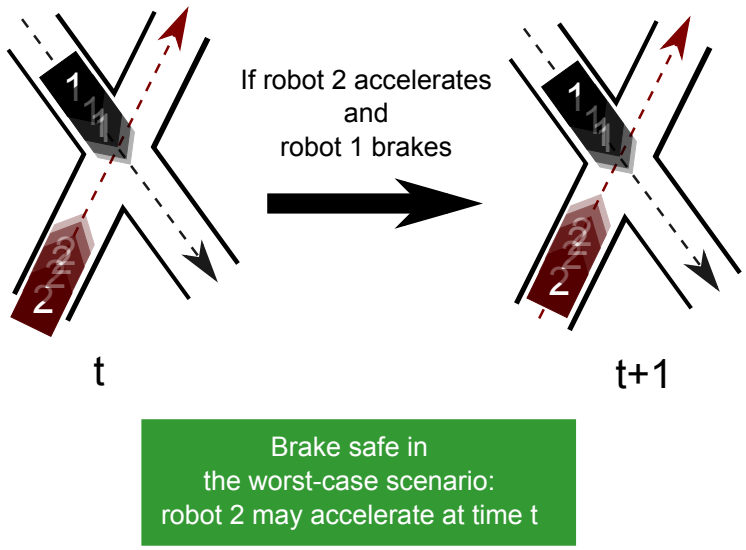


Figure 5.7: In this setting where robot 1 is assumed to have priority, the worst-case scenario under which robot 2 accelerates and robot 1 brakes leads to a new state which is brake safe. In this case, the control law allows robot 2 to accelerate. Robots with lower opacity are used to represent the brake trajectory.

Theorem 5.2.1 (Priority preservation)**.** *Given a priority graph $G \in \mathcal{G}$, the set of brake safe states B_G is positively invariant (in discrete time) for the system under control law g^G, i.e.,*

$$\forall s \in B_G, \forall k \in \mathbb{N}, \phi(k, s, g^G) \in B_G \tag{5.12}$$

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

$$\forall s \in B_G, \forall t \geq 0, \phi_x(t, s, g^G) \in \chi_G^{\text{free}} \tag{5.13}$$

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 (5.12)). Moreover, the flow of the system remains in χ_G^{free} in continuous time (see Equation (5.13)), i.e., no collision occurs and priorities are preserved. It is a direct consequence of Theorem 5.2.2 and appears as a limiting case.

5.2.2 Robustness

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 robots brake as much as possible, i.e., leading to an emergency stop. This property stated in Theorem 5.2.2 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 a loss of sensing/communication abilities.

Theorem 5.2.2 (A broad class of priority preserving controls)**.** *Given a priority graph $G \in \mathcal{G}$, an initial condition $s \in B_G$, and a control $\mathbf{u} \in \mathbf{U}$ that satisfies:*

$$\forall k \in \mathbb{N}, \mathbf{u}(k) \leq g^G(\phi(k, s, \mathbf{u})) \tag{5.14}$$

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

$$\forall k \in \mathbb{N}, \phi(k, s, \mathbf{u}) \in B_G \tag{5.15}$$

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

$$\forall t \geq 0, \phi_x(t, s, \mathbf{u}) \in \chi_G^{\text{free}} \tag{5.16}$$

Proof. Take a priority graph $G \in \mathcal{G}$, an initial condition $s \in B_G$ and a control $\mathbf{u} \in \mathbf{U}$ satisfying Equation 5.14. By induction, it is sufficient to prove that the flow

is collision-free for $t \in [0,1]$ and the reached state $\phi(1,s,\mathbf{u})$ is brake safe. Now, we prove that the flow of Theorem 5.2.2 does not intersect χ_G^{obs} for $t \in [0,1]$. Take arbitrary $t \in [0,1]$: we have to prove that for all $(j,i) \in E(G)$, $\phi_x(t,s,\mathbf{u}) \in \chi_{j\succ i}^{\text{free}}$. By construction of g^G, for each robot i, there are two cases:

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

$$\phi_i(t,s,\mathbf{u}) = \phi_i(t,s,\underline{\mathbf{u}}) \tag{5.17}$$

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

$$\phi_j(t,s,\mathbf{u}) \geq \phi_j(t,s,\underline{\mathbf{u}}) \tag{5.18}$$

  Since s is brake safe, $\phi_x(t,s,\underline{\mathbf{u}}) \in \chi_{j\succ i}^{\text{free}}$. Hence, by Property 2.2.1, Equations (5.17) and (5.18) ensure that $\phi_x(t,s,\mathbf{u}) \in \chi_{j\succ i}^{\text{free}}$ as well.

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

$$\phi_i(t,s,\tilde{\mathbf{u}}^i) = \phi_i(t,s,\overline{\mathbf{u}}) \geq \phi_i(t,s,\mathbf{u}) \tag{5.19}$$

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

$$\phi_j(t,s,\tilde{\mathbf{u}}^i) = \phi_j(t,s,\underline{\mathbf{u}}) \leq \phi_j(t,s,\mathbf{u}) \tag{5.20}$$

  Since $\phi_x(t,s,\tilde{\mathbf{u}}^i) \in \chi_G^{\text{free}}$, $\phi_x(t,s,\tilde{\mathbf{u}}^i) \in \chi_{j\succ i}^{\text{free}}$, and by Property 2.2.1, Equations (5.19) and (5.20) ensure that $\phi_x(t,s,\mathbf{u}) \in \chi_{j\succ i}^{\text{free}}$ as well.

As a final step, we 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 $(j,i) \in E(G)$, $\phi_x(t,s^1,\underline{\mathbf{u}}) \in \chi_{j\succ i}^{\text{free}}$. 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:

$$\phi_i(t,s^1,\underline{\mathbf{u}}) = \phi_i(1+t,s,\underline{\mathbf{u}}) \tag{5.21}$$

  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:

$$\phi_j(t,s^1,\underline{\mathbf{u}}) \geq \phi_j(1+t,s,\underline{\mathbf{u}}) \tag{5.22}$$

  Since s is brake safe, $\phi_x(1+t,s,\underline{\mathbf{u}}) \in \chi_{j\succ i}^{\text{free}}$. Hence, by Property 2.2.1, Equations (5.21) and (5.22) ensure that $\phi_x(t,s^1,\underline{\mathbf{u}}) \in \chi_{j\succ i}^{\text{free}}$ as well.

- $g_i^G(s) = \overline{u}_i$: then, by construction of the control law, $\phi_x(1+t,s,\tilde{\mathbf{u}}^i) \in \chi_G^{\text{free}}$. 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 $\phi_x(1+t,s,\tilde{\mathbf{u}}^i) \in \chi_G^{\text{free}}$, $\phi_x(t,\tilde{s}^1,\underline{\mathbf{u}}) \in \chi_G^{\text{free}}$.

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

$$\tilde{s}_i^1 = \phi_i(1,s,\tilde{\mathbf{u}}^i) = \phi_i(1,s,\overline{\mathbf{u}}) \geq \phi_i(1,s,\mathbf{u}) = s_i^1 \tag{5.23}$$

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

$$\tilde{s}_j^1 = \phi_j(1, s, \tilde{\mathbf{u}}^i) = \phi_j(1, s, \underline{\mathbf{u}}) \leq \phi_j(1, s, \mathbf{u}) = s_j^1 \tag{5.24}$$

Hence, by order-preservation, Equations (5.23) and (5.24) imply:

$$\phi_i(t, \tilde{s}^1, \underline{\mathbf{u}}) \geq \phi_i(t, s^1, \underline{\mathbf{u}}) \tag{5.25}$$
$$\phi_j(t, \tilde{s}^1, \underline{\mathbf{u}}) \leq \phi_j(t, s^1, \underline{\mathbf{u}}) \tag{5.26}$$

Since $\phi_x(t, \tilde{s}^1, \underline{\mathbf{u}}) \in \chi_G^{\text{free}}$, $\phi_x(t, \tilde{s}^1, \underline{\mathbf{u}}) \in \chi_{j \succ i}^{\text{free}}$, and by Property 2.2.1, Equations (5.25) and (5.26) ensure that $\phi_x(t, s^1, \underline{\mathbf{u}}) \in \chi_{j \succ i}^{\text{free}}$ as well.

□

To illustrate the interest of Theorem 5.2.2, given priorities G and an initial condition $s \in B_G$ consider the two examples below.

Example 5.2.1 (Individual brake application)**.** Consider a control $\mathbf{u} \in \mathbf{U}$ satisfying:

$$\forall k \in \mathbb{N}, \mathbf{u}_i(k) = \begin{cases} \underline{u}_i & \text{if } k \in K \\ g_i^G(\phi(k, s, \mathbf{u})) & \text{else.} \end{cases} \tag{5.27}$$
$$\forall j \in \mathcal{R}, j \neq i, \mathbf{u}_j(k) = g_j^G(\phi(k, s, \mathbf{u})) \tag{5.28}$$

$i \in \mathcal{R}$ 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 maneuver is performed instead. The condition of Theorem 5.2.2 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 5.2.2 (Simultaneous brake application)**.** Consider a control $\mathbf{u} \in \mathbf{U}$ satisfying:

$$\forall k \in \mathbb{N}, \mathbf{u}(k) = \begin{cases} \underline{u} & \text{if } k \in K \\ g^G(\phi(k, s, \mathbf{u})) & \text{else.} \end{cases} \tag{5.29}$$

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 5.2.2 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.

5.2.3 Liveness

As in the case of velocity control, we aim at guaranteeing liveness, i.e., the guarantee that every robot $i \in \mathcal{R}$ eventually reaches the region $\chi^{\text{goal}} := \overline{x}^{\text{obs}} + \mathbb{R}^n_+$.

Theorem 5.2.3 (Liveness)**.** *Given an acyclic priority graph G and an initial brake safe state $s \in B_G$, there exists $T > 0$ such that:*

$$\phi_x(T, s, g^G) \in \chi^{\text{goal}} \tag{5.30}$$

Again, the idea of the proof is that under acyclic priorities, there is always a non exited robot able to travel at maximum throttle command until it exits the intersection.

Proof. Take an acyclic priority graph G. Consider the trajectory of the robots under control law g^G. G being acyclic, there exists an extremal vertex $i_1 \in \mathcal{R}$ such that for all $j \in \mathcal{R}$, $(j, i_1) \notin E(G)$. 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 (it will reach position $\overline{x}_i^{\text{obs}}$) 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 $i_{m+1} \in \mathcal{R} \setminus \{i_1 \cdots i_m\}$ such that for all $j \in \mathcal{R} \setminus \{i_1 \cdots i_m\}$, $(j, i_{m+1}) \notin E(G)$. 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 yields a sequence $(T_1 \cdots T_n)$ and all robots have exited the intersection at time $T := T_n$. □

Chapter 6

Robustness with respect to bounded noise

Contents

6.1 Control model with bounded noise 84
6.2 Evolution of the non-deterministic information state 85
6.3 The proposed decentralized control law 87
6.3.1 Priority preservation 87
6.3.2 Robustness 88
6.3.3 Liveness 90

In the plan-as-program approach, a low-level controller is assumed to be able to follow the planned trajectory. Uncertainty is taken into account at the control phase. This is known as the trajectory tracking problem. Many trajectory tracking systems have been proposed for different robot dynamics models [63, 80, 92, 122, 141]. In [131], a linearisation of the robot dynamics model around the tracked trajectory enables to obtain a linear-quadratic regulator [129], and under Gaussian models of uncertainty, the a-priori distribution of the trajectory around the tracked trajectory can be computed. It makes possible to compare several possible motion planning strategies in terms of collision probability, and to select one of them based on some criteria/cost function. However, even when a-priori knowledge on uncertainty is used as in [131] to plan the trajectory, the trajectory tracking approach is still quite decoupled as the reference trajectory remains unchanged as new information comes in during the execution of the plan. It can result in undesirable behaviors particularly in case of large deviation from the reference trajectory.

In priority-based coordination, there is no reference trajectory to track. There are assigned priorities to preserve which is ensured by a control law configured by the assigned priorities (see previous chapters of the present part). In this setting, information on uncertainty can be used as an additional resource to take into account when acting, i.e., as an additional input for the control law. In [75, 77], the information space approach is proposed. The information state at time t contains all the information history up to date t. Under probabilistic uncertainty, the current information state can be considered as the distribution of the current state of the system conditionally to current history. Under non deterministic uncertainty, the current information state is the set of all possible "true" current states of the robot: one can see the non-deterministic information state as a "bubble" of possible

current "true" states. In this approach, the action of robots is a function of the current information state: the control law takes into account the uncertainty on the estimated current state to decide action. The approach has since become standard (see, e.g., [106, 45]). This chapter espouses the information space approach.

Priority preserving control under bounded noise in the coordination space is considered. As in Chapter 5, priorities guide the action of robots by configuring the control law. However, in this chapter, the control law does not take the current state of the robot as input as it is not known. We take a model of bounded uncertainty which enables to apply the non-deterministic information space approach [75, 77]. The control law takes as input the non-deterministic information state of the robots, i.e., the set of all possible current positions and velocities of robots along their paths. Uncertainty is only considered from the coordination space point of view. Uncertainty on the path following assumption (lateral control) and more generally realistic models of uncertainty based on real sensors/actuators models are beyond the ambition of the present thesis (it is mentioned as a perspective in the concluding part). The present chapter only aims at providing elements demonstrating the robustness abilities of the proposed priority preserving approach in the presence of bounded noise.

Sketch of the chapter Section 6.1 exposes the second-order dynamics model with bounded noise. Section 6.2 defines the so-called non-deterministic information state for our particular multi robot system and provides the equations describing its evolution through time. The last section of the chapter builds a priority preserving control law taking into account uncertainty information by considering the worst-case scenario. It is guaranteed that for all possible errors/perturbations in sensing/control, no collision occurs, priorities are respected and all robots eventually go through the intersection. Additionally, the brake safety property of Chapter 5 stating that robots may safely brake at any point of time without violating priorities still holds.

6.1 Control model with bounded noise

We slightly modify the model of Chapter 5 to account for bounded control noise. Each robot i is modeled as a second-order control 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:

$$\dot{x_i}(t) = v_i(t) + \mathbf{1}_{v_i(t)=\overline{v}_i} \mathbf{d}_i^v(t) \tag{6.1}$$

$$\dot{v_i}(t) = (\mathbf{u}_i(t) + \mathbf{d}_i^u(t))\ \delta(\mathbf{u}_i(t) + \mathbf{d}_i^u(t), v_i(t)) \tag{6.2}$$

with the same notations as in Chapter 5 and with $\mathbf{1}_C$ returning 1 if condition C holds, 0 else. Basically, $\mathbf{d}^v$ models the uncertainty on maintaining maximum velocity and $\mathbf{d}^u$ models the uncertainty on the brake command. $\mathbf{d} = (\mathbf{d}^v, \mathbf{d}^u)$ is the overall exogenous control uncertainty signal. We assume that control uncertainty is bounded and we let $D := \prod_{i \in \mathcal{R}} D_i$ with $D_i := [\underline{d}_i, \overline{d}_i]$ and $\underline{d}_i \in \mathbb{R}_-^2$ and $\overline{d}_i \in \mathbb{R}_+^2$.

We let $\mathbf{D}_i$ denote the set of uncertainty controls $\mathbf{d}_i$ taking values in D_i and $\mathbf{D} := \prod_{i\in\mathcal{R}} \mathbf{D}_i$. We let $t \mapsto \phi_i(t, s_i, \mathbf{u}_i, \mathbf{d}_i)$ denote the flow of the system starting at initial condition $s_i \in S_i$ with control $\mathbf{u}_i \in \mathbf{U}_i$ and uncertainty control $\mathbf{d}_i$. As in Chapter 5, projected flows are defined as follows: $\phi_x := \pi_x \circ \phi$ and $\phi_v := \pi_v \circ \phi$. We introduce a partial order for uncertainty signals as follows:

$$\forall \mathbf{d}_i^1, \mathbf{d}_i^2 \in \mathbf{D}_i, \mathbf{d}_i^1 \preceq \mathbf{d}_i^2 \text{ if } \forall t \geq 0, \mathbf{d}_i^1(t) \leq \mathbf{d}_i^2(t) \tag{6.3}$$

Property 6.1.1 (Order preservation)**.** *The flow $t \mapsto \phi_i(t, s_i, \mathbf{u}_i, \mathbf{d}_i)$ is order-preserving with regards to s_i, $\mathbf{u}_i$ and $\mathbf{d}_i$.*

Finally, we make the following assumptions for all $i \in \mathcal{R}$:

$$\underline{d}_i^v + \overline{v}_i > 0 \tag{6.4}$$
$$\overline{u}_i + \underline{d}_i^u > 0 \tag{6.5}$$
$$\underline{u}_i + \overline{d}_i^u < 0 \tag{6.6}$$

Basically, it means that:

- even with uncertainty on maintaining maximum velocity, the velocity is always positive;
- even with uncertainty on control, when a robot applies maximum throttle command, it will effectively accelerate;
- and when a robot applies maximum brake command, it will effectively brake.

6.2 Evolution of the non-deterministic information state

We let 2^A denote the power set of any set A. We assume that we have observations at the beginning of every time slot. We model observations as a signal $\mathbf{y} : \mathbb{R}_+^* \to 2^S$ satisfying $\mathbf{y}(t) = S$ if $t \notin \mathbb{N}$, and $\mathbf{y}(k)$ is a parallelepiped:

$$\forall k \in \mathbb{N}_+^*, \mathbf{y}(k) = \prod_{i\in\mathcal{R}} \mathbf{y}_i(k) = \prod_{i\in\mathcal{R}} \mathbf{y}_i^x(k) \times \mathbf{y}_i^v(k) \tag{6.7}$$

$\mathbf{y}_i^x(k)$ denotes the observation on the position of robot i at time slot k and $\mathbf{y}_i^v(k)$ the observation on the velocity of robot i at time slot k. An observation $\mathbf{y}(k)$ provides a set of possible (true) states given the sensors information. $\mathbf{y}(t) = S$ if $t \notin \mathbb{N}$ means that there is no observation data at time $t \notin \mathbb{N}$. We let $\mathbf{Y}_i$ denote the set of observation signals $\mathbf{y}_i$ satisfying the above assumptions and $\mathbf{Y} := \prod_{i\in\mathcal{R}} \mathbf{Y}_i$.

We remind that the non-deterministic information state at time t provides the set of possible (true) states at time t: if the current non-deterministic information state is $\hat{s} \in 2^S$, the current (true) state $s \in S$ satisfies $s \in \hat{s}$. The evolution of the non-deterministic information state accounts for both the uncertainty on control

and on sensing. Given $k \in \mathbb{N}$, for $t \in (k, k+1)$, the uncertainty on control (through $\mathbf{d}$) increases the size of possible states as time goes by. At time $k+1$, a new observation is available, and the new state of the system necessarily belongs to the range given by the observation.

The above statements lead to the non-deterministic information state flow $t \mapsto \hat{\phi}(t, \hat{s}, \mathbf{u}, \mathbf{y})$ associated to an initial condition $\hat{s}$, a control $\mathbf{u}$ and an observation signal $\mathbf{y} \in \mathbf{Y}$ defined as follows.:

$$\hat{\phi}(0, \hat{s}, \mathbf{u}, \mathbf{y}) := \hat{s} \tag{6.8}$$

$$\forall k \in \mathbb{N}, \forall t \in (0,1), \hat{\phi}(k+t, \hat{s}, \mathbf{u}, \mathbf{y}) := \left\{ \phi(t, s, \mathbf{u}, \mathbf{d}) : \mathbf{d} \in \mathbf{D}, s \in \hat{\phi}(k, \hat{s}, \mathbf{u}, \mathbf{y}) \right\} \tag{6.9}$$

$$\forall k \in \mathbb{N}, \hat{\phi}(k+1, \hat{s}, \mathbf{u}, \mathbf{y}) := \left\{ \phi(1, s, \mathbf{u}, \mathbf{d}) : \mathbf{d} \in \mathbf{D}, s \in \hat{\phi}(k, \hat{s}, \mathbf{u}, \mathbf{y}) \right\} \cap \mathbf{y}(k+1) \tag{6.10}$$

The evolution of the non-deterministic information state flow is illustrated in Figure 6.1. By order-preservation, as D and $y(k)$ are parallelepipeds, and as the intersection of two parallelepipeds is a parallelepiped, it is clear that starting from an initially parallelepipedic non-deterministic information state $\hat{s}$, $\hat{\phi}(t, \hat{s}, \mathbf{u}, \mathbf{y})$ is a parallelepiped at any point of time. Note that it makes its computation easier as only two extremal points need to be computed. As for deterministic flows, projected flows are defined as follows: $\hat{\phi}_x := \pi_x \circ \hat{\phi}$ and $\hat{\phi}_v := \pi_v \circ \hat{\phi}$.

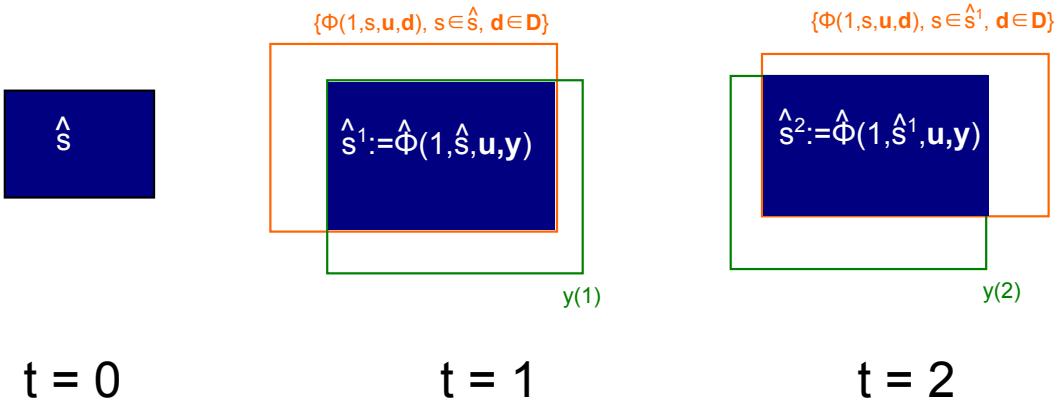


Figure 6.1: Evolution of the non-deterministic information state. For $t \in (0,1)$, the "size" of the non-deterministic information state grows until a new observation is available at $t = 1$. The intersection with $y(1)$ at $t = 1$ enables to take into account the new observation and reduces the "size" of the updated non-deterministic information state $\hat{s}^1$.

We let $\mathbf{S}$ denote the constant observation signal $\mathbf{y}(t) \equiv S$, i.e., there is no observation data for all $t \geq 0$. We define the set of brake safe non-deterministic states $\hat{B}_G$ as follows:

$$\hat{B}_G := \left\{ \hat{s} \in 2^{\chi} : \forall t \geq 0, \hat{\phi}_x(t, \hat{s}, \underline{\mathbf{u}}, \mathbf{S}) \in 2^{\chi_G^{\text{free}}} \right\} \tag{6.11}$$

$$= \left\{ \hat{s} \in 2^{\chi} : \forall s \in \hat{s}, \forall \mathbf{d} \in \mathbf{D}, \forall t \geq 0, \phi_x(t, s, \underline{\mathbf{u}}, \mathbf{d}) \in \chi_G^{\text{free}} \right\} \tag{6.12}$$

A non-deterministic state $\hat{s} \in 2^{\mathcal{X}}$ is brake safe if starting from any true state $s \in \hat{s}$, the flow under maximum brake command $\underline{\mathbf{u}}$ is collision-free for all possible control uncertainty signals $\mathbf{d} \in \mathbf{D}$.

6.3 The proposed decentralized control law

We are going to build a control law for the non-deterministic system. The control law maps the current information state of the system to the control to apply, i.e., it is a map $\hat{g} : 2^S \to U$. We let $t \mapsto \hat{\phi}(t, \hat{s}, \hat{g}, \mathbf{y})$ denote the flow $t \mapsto \hat{\phi}(t, \hat{s}, \mathbf{u}, \mathbf{y})$ where $\mathbf{u}$ satisfies:

$$\forall k \in \mathbb{N}, \mathbf{u}(k) \equiv \hat{g}(\hat{\phi}(k, \hat{s}, \mathbf{u}, \mathbf{y})) \tag{6.13}$$

For all $i \in \mathcal{R}$, we define the operator $\widetilde{\sup}^i$ and the uncertainty signal $\tilde{\mathbf{d}}^i$ as follows:

$$\forall j \in \mathcal{R}, \widetilde{\sup}^i_j(\hat{s}) := \begin{cases} \sup \hat{s}_i & \text{if } j = i \\ \inf \hat{s}_j & \text{if } j \neq i \end{cases} \tag{6.14}$$

$$\forall j \in \mathcal{R}, \tilde{\mathbf{d}}^i_j := \begin{cases} \overline{\mathbf{d}}_i & \text{if } j = i \\ \underline{\mathbf{d}}_j & \text{if } j \neq i \end{cases} \tag{6.15}$$

Basically, $\widetilde{\sup}^i(\hat{s})$ represents the worst-case possible true state of the system for collisions with regards to $\chi^{\text{obs}}_{j \succ i}$. Similarly, $\tilde{\mathbf{d}}^i_j$ is the worst-case possible disturbance with regards to $\chi^{\text{obs}}_{j \succ i}$. The rationale of the proposed control law is to apply the same control law as without uncertainty but considering the worst-case scenario (both worst-case disturbance and worst-case true state). The control law can be formulated synthetically as follows:

$$\hat{g}^G_i(\hat{s}) := \begin{cases} \underline{u}_i & \text{if } \exists (j,i) \in E(G), \exists t \geq 0 \text{ s.t. } \phi_x(t, \widetilde{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \in \chi^{\text{obs}}_{j \succ i} \\ \overline{u}_i & \text{else.} \end{cases} \tag{6.16}$$

6.3.1 Priority preservation

As previously, we first focus on the most important property: safety, i.e., priority preservation.

Theorem 6.3.1 (Priority preservation)**.** *Given a priority graph $G \in \mathcal{G}$, the set of brake safe non-deterministic information states $\hat{B}_G$ is positively invariant (in discrete time) for the non-deterministic system under control law $\hat{g}^G$, i.e.,*

$$\forall \mathbf{y} \in \mathbf{Y}, \forall \hat{s} \in \hat{B}_G, \forall k \in \mathbb{N}, \hat{\phi}(k, \hat{s}, \hat{g}^G, \mathbf{y}) \in \hat{B}_G \tag{6.17}$$

Moreover, the non-deterministic system remains in $2^{\chi_G^{\text{free}}}$ through time, i.e.,

$$\forall \mathbf{y} \in \mathbf{Y}, \forall \hat{s} \in \hat{B}_G, \forall t \geq 0, \hat{\phi}_x(t, \hat{s}, \hat{g}^G, \mathbf{y}) \in 2^{\chi_G^{\text{free}}} \tag{6.18}$$

As in Chapter 5, the above theorem is a limit case of Theorem 6.3.2 proved in the sequel.

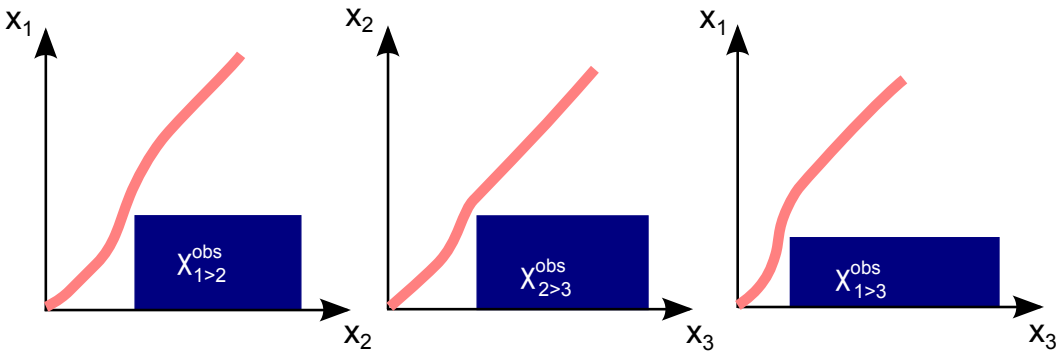


Figure 6.2: Look of the trajectory for a three-robot system with acyclic assigned priorities $1 \succ 2$, $2 \succ 3$ and $1 \succ 3$ under control law g^G. A band is used to represent the set of all possible real configurations through time.

6.3.2 Robustness

As in Chapter 5, the control law $\hat{g}_i^G$ returns the maximum control value that robot i can safely apply, but it is in fact always safe to apply a lower control value. Hence, we obtain the same robustness property that is highly valuable for applications in autonomous cars.

Theorem 6.3.2 (A broad class of priority preserving controls)**.** *Given a priority graph $G \in \mathcal{G}$, an initial condition $\hat{s} \in \hat{B}_G$, an observation signal $\mathbf{y} \in \mathbf{Y}$ and a control $\mathbf{u} \in \mathbf{U}$ that satisfies:*

$$\forall k \in \mathbb{N}, \mathbf{u}(k) \leq \hat{g}^G(\hat{\phi}(k, \hat{s}, \mathbf{u}, \mathbf{y})) \tag{6.19}$$

The set of non-deterministic brake safe states $\hat{B}_G$ is positively invariant (in discrete time), i.e.,

$$\forall k \in \mathbb{N}, \hat{\phi}(k, \hat{s}, \mathbf{u}, \mathbf{y}) \in \hat{B}_G \tag{6.20}$$

Moreover, the configuration of the non-deterministic system remains in $2^{\chi_G^{\text{free}}}$ through time, i.e.,

$$\forall t \geq 0, \hat{\phi}_x(t, \hat{s}, \mathbf{u}, \mathbf{y}) \in 2^{\chi_G^{\text{free}}} \tag{6.21}$$

Proof. Take a priority graph $G \in \mathcal{G}$, an initial condition $\hat{s} \in \hat{B}_G$ and a control $\mathbf{u} \in \mathbf{U}$ satisfying Equation (6.19). By induction, it is sufficient to prove that the flow remains in $2^{\chi_G^{\text{free}}}$ for $t \in [0, 1]$ and the reached state $\hat{\phi}(1, \hat{s}, \mathbf{u}, \mathbf{y}) \in \hat{B}_G$. First, we prove that the flow of Theorem 6.3.2 does not intersect $2^{\chi_G^{\text{obs}}}$ for $t \in [0, 1]$. Take arbitrary $t \in [0, 1]$ and $(j, i) \in E(G)$: we have to prove that $\hat{\phi}_x(t, \hat{s}, \mathbf{u}, \mathbf{y}) \in 2^{\chi_{j \succ i}^{\text{free}}}$. By construction of $\hat{\phi}$, it is equivalent to prove that for all $\mathbf{d} \in \mathbf{D}$ and $s \in \hat{s}$, $\phi_x(t, s, \mathbf{u}, \mathbf{d}) \in \chi_{j \succ i}^{\text{free}}$. By construction of $\hat{g}^G$, there are two cases:

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

  $$\phi_i(t, s, \mathbf{u}, \mathbf{d}) = \phi_i(t, s, \underline{\mathbf{u}}, \mathbf{d}) \tag{6.22}$$

  and by order-preservation:

  $$\phi_j(t, s, \mathbf{u}, \mathbf{d}) \geq \phi_j(t, s, \underline{\mathbf{u}}, \mathbf{d}) \tag{6.23}$$

  Since $\hat{s} \in \hat{B}_G$, $\phi_x(t, s, \underline{\mathbf{u}}, \mathbf{d}) \in \chi_{j \succ i}^{\text{free}}$. Hence, by Property 2.2.1, Equations (6.22) and (6.23) ensure that $\phi_x(t, s, \mathbf{u}, \mathbf{d}) \in \chi_{j \succ i}^{\text{free}}$ as well.

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

  $$\phi_i(t, \widetilde{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) = \phi_i(t, \sup \hat{s}, \overline{\mathbf{u}}, \overline{\mathbf{d}}) \geq \phi_i(t, s, \mathbf{u}, \mathbf{d}) \tag{6.24}$$

  Again, by order-preservation, using $\tilde{\mathbf{u}}_j^i(0) = \underline{u}_j$, $\tilde{\mathbf{d}}_j^i = \underline{\mathbf{d}}_j$ and $\widetilde{\sup}_j^i(\hat{s}) = \inf \hat{s}_j$, we have:

  $$\phi_j(t, \widetilde{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) = \phi_j(t, \inf \hat{s}, \underline{\mathbf{u}}, \underline{\mathbf{d}}) \leq \phi_j(t, s, \mathbf{u}, \mathbf{d}) \tag{6.25}$$

  Since $\phi_x(t, \widetilde{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \in \chi_{j \succ i}^{\text{free}}$, by Property 2.2.1, Equations (6.24) and (6.25) ensure that $\phi_x(t, s, \mathbf{u}, \mathbf{d}) \in \chi_{j \succ i}^{\text{free}}$.

As a final step, we prove that the reached state $\hat{\phi}(1, \hat{s}, \mathbf{u}, \mathbf{y})$ is brake safe. It is sufficient to prove that:

$$\hat{s}^1 := \{\phi(1, s, \mathbf{u}, \mathbf{d}) : \mathbf{d} \in \mathbf{D}, s \in \hat{s}\} \in \hat{B}_G \tag{6.26}$$

$\hat{\phi}(1, \hat{s}, \mathbf{u}, \mathbf{y})$ is indeed a subset of $\hat{s}^1$ by construction of $\hat{\phi}$ (see Equation (6.10)). Take arbitrary $t \geq 0$ and $(j, i) \in E(G)$: we have to prove that for all $s^1 \in \hat{s}^1$ and $\mathbf{d}^1 \in \mathbf{D}$, we have $\phi_x(t, s^1, \underline{\mathbf{u}}, \mathbf{d}^1) \in \chi^{\text{free}}_{j \succ i}$.

Take arbitrary $s^1 \in \hat{s}^1$ and $\mathbf{d}^1 \in \mathbf{D}$. We have $s^1 = \phi(1, s, \mathbf{u}, \mathbf{d})$ with $\mathbf{d} \in \mathbf{D}$ and $s \in \hat{s}$. Consider $\mathbf{d}^2 \in \mathbf{D}$ and $\mathbf{u}^2 \in \mathbf{U}$ satisfying $\mathbf{d}^2(0) = \mathbf{d}(0)$, $\mathbf{u}^2(0) = \mathbf{u}(0)$ and for all $k \in \mathbb{N}$, $\mathbf{d}^2(k+1) = \mathbf{d}^1(k)$ and $\mathbf{u}^2(k+1) = \underline{u}$. By construction, we have: $\phi(t, s^1, \underline{\mathbf{u}}, \mathbf{d}^1) = \phi(1 + t, s, \mathbf{u}^2, \mathbf{d}^2)$.

As a result, we have to prove that $\phi_x(1 + t, s, \mathbf{u}^2, \mathbf{d}^2) \in \chi^{\text{free}}_{j \succ i}$. As previously, there are two cases:

- $g^G_i(s) = \underline{u}_i$: then, we have $\mathbf{u}^2_i = \underline{\mathbf{u}}_i$, so that:

  $$\phi_i(1 + t, s, \mathbf{u}^2, \mathbf{d}^2) = \phi_i(1 + t, s, \underline{\mathbf{u}}, \mathbf{d}^2) \tag{6.27}$$

  Moreover, by order-preservation, we have:

  $$\phi_j(1 + t, s, \mathbf{u}^2, \mathbf{d}^2) \geq \phi_j(1 + t, s, \underline{\mathbf{u}}, \mathbf{d}^2) \tag{6.28}$$

  Since s is brake safe, $\phi_x(1 + t, s, \underline{\mathbf{u}}, \mathbf{d}^2) \in \chi^{\text{free}}_{j \succ i}$. Hence, by Property 2.2.1, Equations (6.27) and (6.28) ensure that $\phi_x(1 + t, s, \mathbf{u}^2, \mathbf{d}^2) \in \chi^{\text{free}}_{j \succ i}$ as well.

- $g^G_i(s) = \overline{u}_i$: then by construction of the control law, $\phi_x(1+t, \widetilde{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \in \chi^{\text{free}}_G$. Using $\mathbf{u}^2_i \leq \tilde{\mathbf{u}}^i_i$, $\widetilde{\sup}^i_i(\hat{s}) = \sup \hat{s}_i$ and $\tilde{\mathbf{d}}^i_i = \overline{\mathbf{d}}_i$, by order-preservation, we have:

  $$\phi_i(1 + t, \widetilde{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) = \phi_i(1 + t, \sup \hat{s}, \tilde{\mathbf{u}}^i, \overline{\mathbf{d}}) \geq \phi_i(1 + t, s, \mathbf{u}^2, \mathbf{d}^2) \tag{6.29}$$

  Moreover, by order preservation, using $\tilde{\mathbf{u}}^i_j = \underline{\mathbf{u}}_j$, $\widetilde{\sup}^i_j(\hat{s}) = \inf \hat{s}_j$ and $\tilde{\mathbf{d}}^i_j = \underline{\mathbf{d}}_j$, we have:

  $$\phi_j(1 + t, \widetilde{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) = \phi_j(1 + t, \inf \hat{s}, \underline{\mathbf{u}}, \underline{\mathbf{d}}) \leq \phi_j(1 + t, s, \mathbf{u}^2, \mathbf{d}^2) \tag{6.30}$$

  Since $\phi_x(1 + t, \widetilde{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \in \chi^{\text{free}}_{j \succ i}$, by Property 2.2.1, Equations (6.29) and (6.30) ensure that $\phi_x(1 + t, s, \mathbf{u}^2, \mathbf{d}^2) \in \chi^{\text{free}}_{j \succ i}$ as well.

□

6.3.3 Liveness

Despite uncertainty, the proposed control still ensures all robots will eventually go through the intersection. As in previous chapters, robots are expected to eventually reach the region $\chi^{\text{goal}} := \overline{x}^{\text{obs}} + \mathbb{R}^n_+$.

Theorem 6.3.3 (Liveness)**.** *Given an acyclic priority graph G, an initial brake safe non-deterministic state $\hat{s} \in \hat{B}_G$ and an observation signal $\mathbf{y} \in \mathbf{Y}$, there exists $T > 0$ such that:*

$$\hat{\phi}_x(T, \hat{s}, \hat{g}^G, \mathbf{y}) \in 2^{\chi^{\text{goal}}} \tag{6.31}$$

A proof of the above theorem is provided in Appendix B.3 (under weaker assumptions).

Conclusions

Part I suggested to use priorities as a plan to guide robots. In traditional planning, the plan is a reference trajectory to track and the trajectory tracking problem is a well-established problem with many existing solutions consisting in devising a control law configured by the reference trajectory in charge of tracking. However, in priority-based coordination, there is no reference trajectory. The plan is the priority graph and there is no standard tool to control robots under assigned priorities. Devising such tools has been the topic of the present part. The proposed control laws are configured by the assigned priorities, and guarantee priority preservation and liveness (all robots eventually go through the intersection). First of all, as priorities are assigned, the combinatorial complexity of multi robot control (see [25]) is avoided, and computing the output of the control laws proposed in this part is of polynomial complexity. Moreover, in contrast with a trajectory tracking approach, the priority preservation approach retains some freedom of action at the control phase, as there is a large class of trajectories respecting the assigned priorities (instead of only one reference trajectory). In particular, the proposed control law ensures that robots – only one, several, or even all – may brake at any time without violating priorities. In Chapter 4, we have proposed priority preserving control for robots controlled in velocity. The trajectory resulting from the application of the proposed control law is optimal for the assigned priorities, recovering the existence of a left-greedy optimal trajectory in a given homotopy class noticed in [47]. Chapter 5 demonstrates that the presence of inertia can be handled using the notion of brake safety that merely consists in some kind of anticipation. The proposed control law is decentralized and demonstrates a remarkable robustness regarding unexpected deceleration of robots. The final chapter of the present part has given some elements to take into account uncertainty in priority-based coordination. Under bounded uncertainty, the idea is to consider the worst-case scenario which is well defined when priorities are assigned. Priority preservation and liveness can still be guaranteed as in the deterministic case. This chapter demonstrates the ability of the priority-based approach to handle uncertainty in a reactive manner. For example, if the current uncertainty on the position of robots is very large, under priority preserving control, all robots will brake and eventually stop safely, and will not restart until a sufficiently small uncertainty enables to go safely through the intersection. By contrast, tracking a reference planned trajectory when the uncertainty on position is very large would likely result in collisions. Hence, with a plan execution approach, if the uncertainty becomes very large, the designer should anticipate by providing an emergency maneuver to execute. Then, a new planning phase should be carried out before restarting. With priority-based coordination, such change in uncertainty – even a complete lost of sensing capabilities – can be handled in a reactive manner.

Part III

Priority-based coordination

Introduction

Part I suggested using priorities to guide robots and Part II provided solutions to control robots under assigned priorities. To this point, many aspects of the design of a coordination system at intersections have been left behind. Most importantly, the multi robot coordination system is an open system as robots arrive and exit the intersection through time. Hence, priorities need to be assigned dynamically. Moreover, priority assignment and control under assigned priorities need to be executed in parallel. This part has a more engineering flavor, it specifies the system architecture, how priority assignment and priority preserving control are integrated and how they interface. The proposed approach is inspired from drivers' behavior at signalized intersections. Before entering the intersection, the driver follows the preceding vehicles without colliding, and as long as the traffic signal does not give him/her the right of way, the driver does not go through the intersection. Once the vehicle is given the right of way (green signal), the driver goes through the intersection. However, the driver still retains some reactive abilities and will hopefully not enter the intersection if other vehicles are blocked and/or a pedestrian crosses the road.

In priority-based coordination, the so-called control area is a region of space that robots should not enter unless they have been accepted and assigned a priority with respect to other accepted robots. We adopt a three-layer architecture [44], particularly adapted to the approach considering plans as a resource to guide action. The reactive quality of the system is ensured by a behavior-based layer. Robots' behaviors include 'follow geometric path', 'move forward', 'do not enter the control area', 'respect priorities', 'avoid pedestrians'. The entry of the control area is managed by a central agent, the intersection controller. The intersection controller assigns priorities, yet it does not assign a precise trajectory for the accepted robots. It constitutes the deliberative layer of the system, processing time-consuming tasks reasoning about the future. Finally, robots have a sequencing layer in charge of activating/deactivating/configuring behaviors. The robustness property of the control law ensuring robots may safely brake at any point of time is shown to be of high interest in the proposed architecture. It is indeed possible for, e.g., behavior 'avoid pedestrians' to require the robot to brake to avoid a detected pedestrian, without conflicting with behavior 'respect priorities', as the control law ensures that it is always priority preserving to brake at any point of time (see Theorem 5.2.2). Preliminaries of the presented results can be found in our article [49].

Sketch of the part Chapter 7 describes the system architecture and how priorities may be assigned. Chapter 8 provides simulation results demonstrating safety and robustness of priority-based coordination.

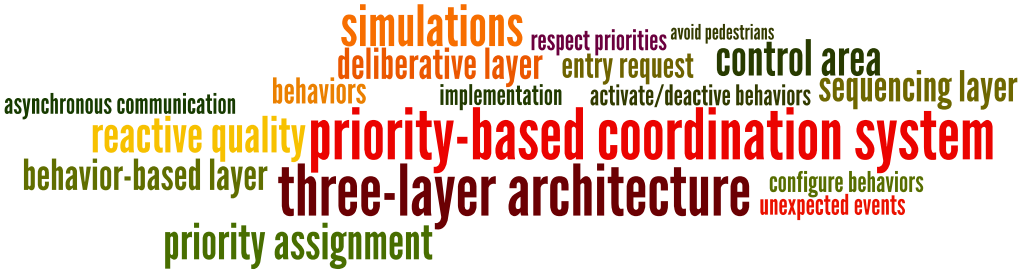

simulations
respect priorities
avoid pedestrians
deliberative layer
entry request
control area
behaviors
implementation
activate/deactive behaviors
sequencing layer
asynchronous communication
reactive quality
priority-based coordination system
behavior-based layer
three-layer architecture
configure behaviors
unexpected events
priority assignment

Table of Contents

7 Overall priority-based coordination system **101**

7.1 Three-layer architecture 101

7.1.1 The intersection controller 101

7.1.2 The behavior-based layer 103

7.1.3 The sequencer 104

7.2 Priority assignment 105

7.2.1 A simple priority assignment policy 105

7.2.2 Request processing liveness 107

7.2.3 Stability guarantees 107

8 Simulations **109**

8.1 Implementation aspects 109

8.2 Simulation results 111

8.2.1 Simulations under deterministic control 112

8.2.2 Robustness regarding unexpected deceleration 113

8.2.3 Robustness regarding bounded uncertainty 114

8.2.4 Stability guarantees under back-pressure control 115

Chapter 7

Overall priority-based coordination system

Contents

7.1 Three-layer architecture 101
7.1.1 The intersection controller 101
7.1.2 The behavior-based layer 103
7.1.3 The sequencer 104
7.2 Priority assignment 105
7.2.1 A simple priority assignment policy 105
7.2.2 Request processing liveness 107
7.2.3 Stability guarantees 107

Sketch of the chapter The first section presents the proposed three-layer architecture and provides details on how layers interact. The second chapter focuses on priority assignment: a simple and easily implementable priority assignment policy is described, and some adaptations in order to guarantee request processing liveness and queues stability are discussed.

7.1 Three-layer architecture

For its ability to design systems with reactive qualities yet retaining planning capabilities, a three-layer architecture is proposed. As noticed in [44], such an architecture organizes control algorithms according to whether their internal state reflects the present, the past, or predictions of the future. Figure 7.1 gives a quick overview of the proposed architecture detailed in the sequel.

7.1.1 The intersection controller

The intersection controller constitutes the deliberative layer of the proposed architecture – reasoning on the future – and manages the control area, defined as a subset of the two-dimensional real space in which the collision area wholly resides. The control area must contain, at least, the subset of the two-dimensional space corresponding to all possible collisions between robots, excluding only regions where collision avoidance is reduced to safe car following (see Figure 7.2). For each

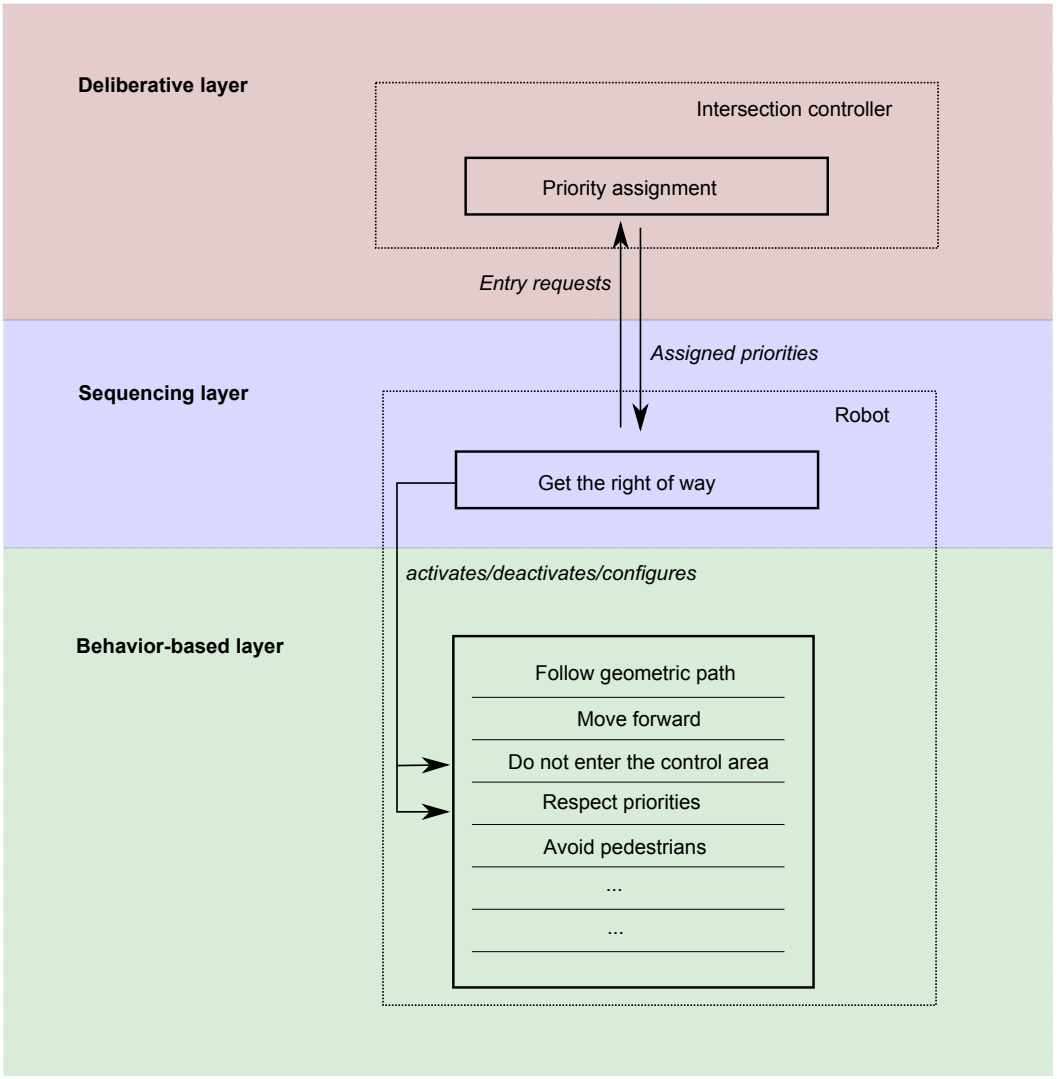


Figure 7.1: The three-layer architecture of priority-based coordination

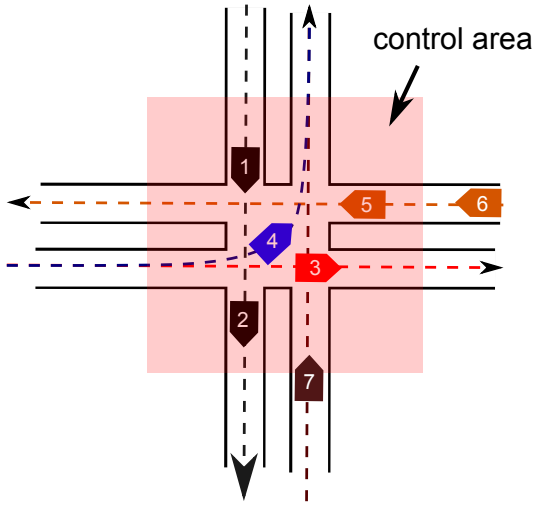


Figure 7.2: The control area.

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 $x_i \in [x_i^{\text{entry}}, x_i^{\text{exit}}]$. To enter the control area, robots send a request to the intersection controller. The job of the intersection controller is to process requests. Either the request is rejected, the robot cannot enter the control area and will have to send a new request; or, the request is accepted and priorities with respect to robots already accepted in the control area are assigned. This task, referred as priority assignment is the topic of Section 7.2.

7.1.2 The behavior-based layer

The behavior-based layer ensures the reactive quality of the system and is implemented by designing control laws.

Move forward All robots implement behavior 'move forward'. Using the second-order dynamics model of Chapter 5, 'move forward' behavior for robot i consists in applying maximum throttle command $\overline{u}_i$. However, this behavior is subsumed [21] by all other behaviors, e.g., the robot will brake if another behavior like 'respect priorities' requires braking.

Do not enter the control area Robots also implement 'do not enter the control area' behavior. 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 and if it is not accepted into the control area, robot i must brake. To formulate this mathematically, the final (and maximal) position reached by robot i

with initial state s_i under impulse control is computed as follows:

$$x_i^{\text{stop}}(s_i) := \max \phi_{x,i}(\mathbb{R}_+, s_i, \mathbf{u}_i^{\text{impulse}}) \tag{7.1}$$

The condition for 'do not enter the control area' behavior to make robot brake then simply becomes: $x_i^{\text{stop}}(s_i) > x_i^{\text{entry}}$.

Respect priorities Robots also implement priority preserving control of Part II (Chapter 5) through 'respect priorities' behavior. This behavior cannot be active before priorities are assigned as it needs priorities as input to configure the control law g^G by specifying the priority graph G.

Additional behaviors In addition to the behaviors presented above, in charge of coordination, robots may implement other behaviors, not directly related to coordination. First of all, the fixed paths assumption (see Section 2.1, Figure 2.1) requires robots to implement a 'follow geometric path' behavior using lateral control. More interestingly, robots may implement behaviors to react to unexpected events. For example, for an application in autonomous vehicles, an 'avoid pedestrians' behavior is a must. It is not conceivable to let autonomous vehicles go through an intersection in an urban area, executing an open-loop planned trajectory without implementing a behavior to detect pedestrians and react accordingly. The benefit of the proposed behavior-based architecture is that a behavior like 'avoid pedestrians' can be implemented in a manner that it subsumes all other behaviors. Most of the time, such reactive safety behaviors will require the robot to brake, and priorities will be conserved as the control law of Chapter 5 guarantees that it is always priority preserving to brake at any point of time (see Theorem 5.2.2). Hence, priority-based coordination can handle a large class of unexpected events – all events requiring one or more robots to brake – without need to replan, i.e., without need to reassign priorities.

7.1.3 The sequencer

In a three-layer architecture, the sequencer's job is to activate/deactivate and/or configure the behaviors [44] that we just listed above. When should 'do not enter the control area' behavior be deactivated in favor of 'respect priorities' behavior ? Note that the state of the sequencer reflects the past as it is necessary to store whether the robot is accepted or not into the control area and to store priorities as well, in order to configure and activate/deactivate behaviors accordingly.

The sequencer communicates with the deliberative layer, i.e., the intersection controller, by sending queries. The goal of these queries is to 'get the right of way'. The condition $x_i^{\text{stop}}(s_i) > x_i^{\text{entry}} - \delta$ is used as the condition to request the entry of the control area. The margin $\delta \geq 0$ enables to anticipate the entry of the control area, so that the intersection controller can possibly accept the robot into the control area in the remaining time, before 'do not enter the control area' behavior's brake condition $x_i^{\text{stop}}(s_i) > x_i^{\text{entry}}$ holds. The sequencer communicates

asynchronously with the intersection controller to ensure a reactive quality. As long as the intersection controller does not accept the robot, the sequencer keeps 'do not enter the control area' behavior active. When the robot is accepted into the control area, 'do not enter the control area' behavior is deactivated in favor of 'respect priorities'. The assigned priorities received by the sequencer in the response of the intersection controller serve as input of 'respect priorities' behavior to configure the control law by specifying the priority graph G.

7.2 Priority assignment

This section focuses on how priorities are assigned, i.e., how the intersection controller processes entry requests.

Priorities as a byproduct of traditional trajectory planning algorithms First of all, it is key to notice that priorities can be obtained as a byproduct of all existing trajectory planning algorithms espousing the plan-as-program paradigm. One can simply assign the priorities induced by the feasible path returned by the planning algorithm. For certain existing algorithms, e.g., in [7], priorities are even directly accessible (in [7], they can be retrieved through the binary variables of the MILP formulation of the problem). Therefore, priority assignment is not the core of the present thesis and we will not provide complex priority assignment policies adapting existing algorithms. In this section, a simple priority assignment policy is proposed resulting in acylic and thus necessarily feasible priorities. Then, perspectives towards "liveness" and "stability" guarantees are presented.

7.2.1 A simple priority assignment policy

The idea of the proposed policy is to let robots spend as little time as possible in the intersection area, inspired from [35]. 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 the newly 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 5.2.3). This can be formulated as follows and implementation aspects are presented in Section 8.1.

First of all, recall the control law of Chapter 5 when robots are controlled in acceleration:

$$g_i^G(s) := \begin{cases} \underline{u}_i & \text{if } \exists (j,i) \in E(G), \exists t \geq 0 \text{ s.t. } \phi_x(t, s, \tilde{\mathbf{u}}^i) \in \chi^{\text{obs}}_{j \succ i} \\ \overline{u}_i & \text{else.} \end{cases} \tag{7.2}$$

As $\kappa_{j \succ i}$ is the cross-section of $\chi^{\text{obs}}_{j \succ i}$, using the definition of $\tilde{\mathbf{u}}^i$, the control law can

be formulated as follows:

$$g_i^G(s) := \begin{cases} \underline{u}_i & \text{if } \exists (j,i) \in E(G), \exists t \geq 0 : (\phi_{x,j}(t, s_j, \underline{\mathbf{u}}_j), \phi_{x,i}(t, s_i, \mathbf{u}_i^{\text{impulse}})) \in \kappa_{j \succ i} \\ \overline{u}_i & \text{else.} \end{cases} \tag{7.3}$$

Consider a robot i that requests the entry of the control area. To decide to accept it or not, we can simulate a trajectory that consists in 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, i.e., following control law g^G. Let $s = (s_j)_{j \in \mathcal{R}}$ denote the current state of robots $j \in \mathcal{R}$, let s_i denote the current state of the requesting robot i and let ς denote the simulated trajectory defined as follows:

$$\forall t \geq 0, \varsigma_i(t) := \phi_i(t, s_i, \overline{\mathbf{u}}_i) \tag{7.4}$$

$$\forall j \in \mathcal{R}, \forall t \geq 0, \varsigma_j(t) := \phi_j(t, s, g^G) \tag{7.5}$$

Then, there are two options:

- if for all $k \in \mathbb{N}$ and for all $j \in \mathcal{R}$ satisfying $\kappa_{ij} \neq \emptyset$, we have:

$$\forall t \geq 0, (\phi_{x,j}(t, \varsigma_j(k), \underline{\mathbf{u}}_j), \phi_{x,i}(t, \varsigma_i(k), \mathbf{u}_i^{\text{impulse}})) \notin \kappa_{j \succ i} \tag{7.6}$$

  the request is accepted and we do:

$$\mathcal{R} \leftarrow \mathcal{R} \cup \{i\} \tag{7.7}$$

$$E(G) \leftarrow E(G) \cup \{(j,i) : j \in \mathcal{R}, \kappa_{ij} \neq \emptyset\} \tag{7.8}$$

- else the request is rejected.

Note that the described algorithm ensures the priority relation to be a partial order, that is G to be a directed acyclic graph at all times. 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 after all already accepted robots. Condition (7.6) ensures that once robot i is accepted and controlled by the control law g^G, if all robots follows g^G (no uncertainty, no unexpected event), the control law will always return $\overline{u}_i$. This means that, in the absence of uncertainty, the coordination system will result in robots either waiting at the entry of the control area (possibly stopped at the entry), or accepted into the control area and applying maximum throttle command, thus going through the intersection at maximum speed. This is what is observable in the simulations of Subsection 8.2.1. However, remind that a key motivation for our priority-based approach is precisely to handle uncertainty. Hence, if some robot does not apply maximum throttle command at some point for an arbitrary reason, the priority preserving control law will ensure that priorities are nevertheless respected as demonstrated by the simulations of Subsection 8.2.2.

7.2.2 Request processing liveness

The weakness of the policy presented above is quite similar to the one of the First-Come-First-Serve policy of [35]. As highlighted in [11] for First-Come-First-Serve reservation policy, and it also holds for the priority assignment policy presented above, handling requests separately and not taking into account the history of requests, causes undesired behaviors like a vehicle in an alley waiting indefinitely at the entry of the intersection.

A solution is presented in [11] to avoid this phenomenon. A batch policy with locking is proposed, consisting of mapping requests to a real value computed using a cost function of the form $f(wait) := a \times wait^b$ where a, b are constants and $wait$ is the estimated amount of the time the robot has been waiting to enter the intersection. The "locking" mechanism is described as follows: when a request r has an associated cost greater than a threshold, then requests from other robots whose path intersects the path of the robot of r will not be granted, until the robot of r is accepted. Interestingly, the proposed policy provably guarantees liveness, i.e., every robot waiting to enter the intersection can eventually enter. This liveness property is different from the one proved in Part II which ensures that once robots are accepted into the intersection, respecting the assigned priorities, they will eventually go through the intersection. The "locking" mechanism can be easily adapted to enhance performance and ensure liveness of the simple priority assignment proposed in Subsection 7.2.1.

7.2.3 Stability guarantees

In traffic signal control, queue lengths are a standard indicator of a control policy's performance. In particular, recently, back-pressure control [127] applied to traffic signals (see, e.g., [134, 140, 52, 51]) aims at providing stability guarantees of the control policy. Loosely speaking, stable queues do not grow indefinitely through time. We believe that this work can be used to endow the priority assignment policy with stability guarantees.

We do not aim to formalize the proposed approach in the general case as it is beyond the scope of the present thesis, so the approach is presented for a particular example. Consider the intersection depicted in Figure 7.3. We let Q_{11}, Q_{12}, Q_{21}, Q_{22} denote the queues lengths at the entry of each path. If the intersection was controlled by a traffic signal, there would be two phases: p_1 and p_2. Assuming this intersection is isolated, under standard back-pressure control, phase p_1 (resp. p_2) is applied if $Q_{11} + Q_{12} \geq Q_{21} + Q_{22}$ (resp. $Q_{11} + Q_{12} < Q_{21} + Q_{22}$).

Traffic signal control is not efficient at low traffic density because of the phase duration. A typical situation is a single vehicle waiting at the intersection for the right of way. The vehicle needs to wait for the end of the current phase before obtaining the right of way. At high traffic density, there are always queues at the entry of the intersection, so traffic signal control is particularly efficient as it lets vehicles move in platoons. Moreover, under back-pressure control, optimal stability

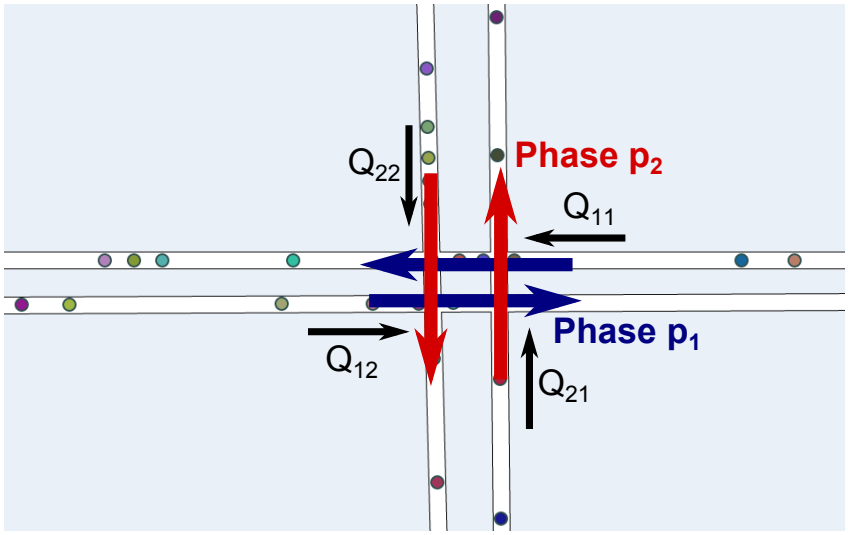


Figure 7.3: A four-path intersection, with two phases and four queues. Phase p_1 empties queues Q_{11} and Q_{12} and phase p_2 empties queues Q_{21} and Q_{22}.

can be proved, i.e., the queuing network is stabilized for all arrival rates that can be stably handled considering all control policies. That is why we propose an adaptive priority assignment policy that consists in applying back-pressure priority assignment if robots accumulate at the entry of the intersection (above a certain threshold), while the basic priority assignment policy presented in Subsection 7.2.1 is applied otherwise. More precisely, a phase duration T and a threshold $\Delta Q^{\text{lim}} \geq 0$ are chosen, and periodically, for $t = 0, T, 2T \cdots$, the phase update algorithm proceeds as follows:

- if $(Q_{11} + Q_{12}) - (Q_{21} + Q_{22}) \geq \Delta Q^{\text{lim}}$: phase p_1 is applied. It means that for all the duration of the phase (T), only the entry requests of robots on the corresponding paths will be accepted. However, requests still need to be accepted according to the priority assignment policy presented in Section 7.2. Typically, when there is a phase switch, the first requests will probably be rejected as there are still robots of the other phase in the intersection. These first requests which are rejected can be seen as a kind of yellow time.
- if $(Q_{21} + Q_{22}) - (Q_{11} + Q_{12}) > \Delta Q^{\text{lim}}$: it is the symmetric case, and phase p_2 is applied.
- otherwise, all phases are activated (both p_1 and p_2), so that the requests of all robots can be potentially accepted. The priority assignment policy is not affected by the phase, and is exactly as presented in Subsection 7.2.1.

Chapter 8

Simulations

Contents

8.1 Implementation aspects 109
8.2 Simulation results 111
8.2.1 Simulations under deterministic control 112
8.2.2 Robustness regarding unexpected deceleration 113
8.2.3 Robustness regarding bounded uncertainty 114
8.2.4 Stability guarantees under back-pressure control 115

Sketch of the chapter The first section provides some details on the implementation of priority-based coordination in simulations. In particular, collision checking, robots random generation and the size of the control area are discussed. Mainly qualitative simulations results are then presented and interpreted.

8.1 Implementation aspects

For the sake of the simplicity, we have implemented our algorithms for circle-shaped robots along straight paths. This choice eases the computation of the obstacle region as every κ_{ij} is the interior of an ellipse whose equation can be easily derived from the radius of robots and the angle between the two straight geometric paths. All robots are supposed to be circle-shaped with a common diameter D. Note that the collision region between each couple of paths can be precomputed once and for all during the design phase of the intersection controller. The lateral control is not simulated and all robots are assumed to follow their assigned geometric path.

To check whether a trajectory is collision-free, as all we can do is to compute a discrete sequence of points, we have used a conservative collision checking algorithm. Basically, to check whether a given flow $(\phi(t, s, \mathbf{u}))_{t\geq 0}$ is collision-free with regards to $\chi^{\text{obs}}_{i\succ j}$, we compute the following sequence of points of $\mathbb{R}^2$:

$$x_i(k) := \phi_{x,i}(k, s, \mathbf{u}) \tag{8.1}$$
$$x_j(k) := \phi_{x,j}(k+1, s, \mathbf{u}) \tag{8.2}$$

Our collision checking algorithm asserts that the flow $(\phi(t, s, \mathbf{u}))_{t\geq 0}$ is collision-free with regards to $\chi^{\text{obs}}_{i\succ j}$ if and only if the sequence $(x_i(k), x_j(k))_{k\in\mathbb{N}}$ is collision-free with regards to $\kappa_{i\succ j}$. This method is illustrated in Figure 8.1. It is direct that it is

conservative as some collision-free trajectories are not asserted to be collision-free; yet the difference vanishes for small enough time slot length.

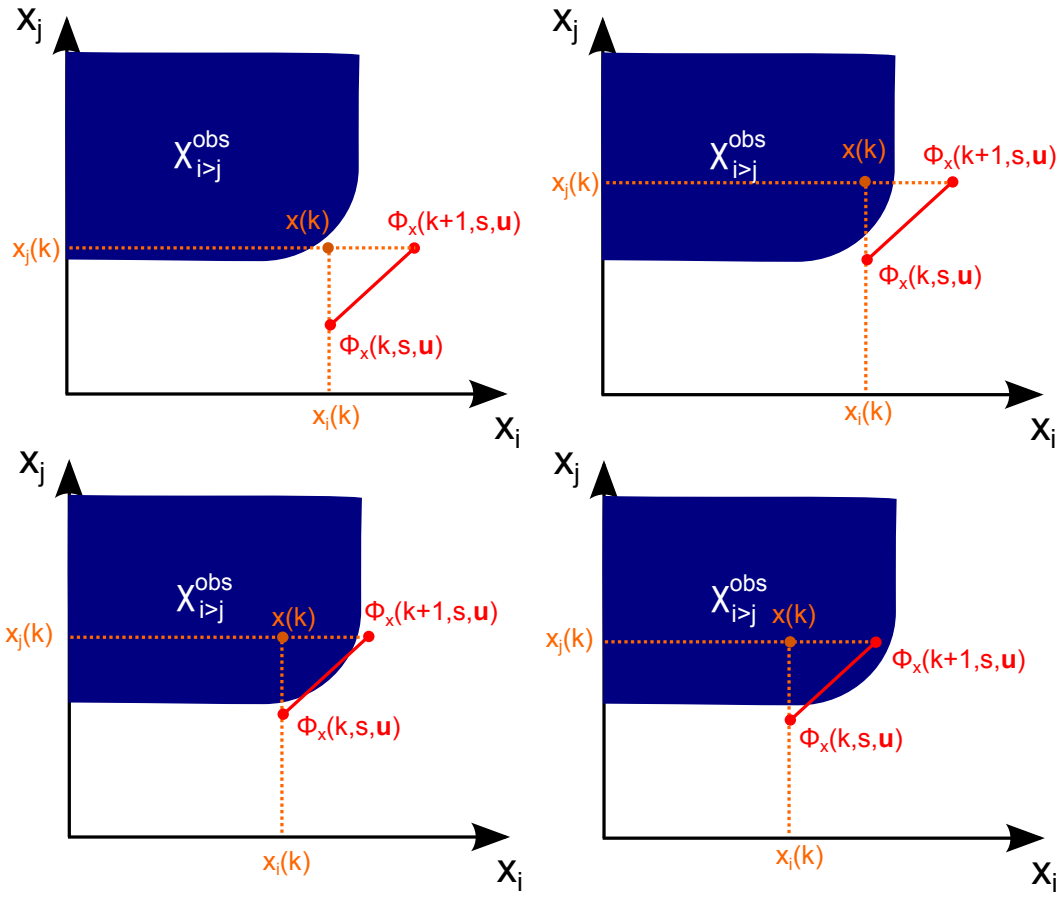


Figure 8.1: Method used for collision checking based on a discrete sequence of points. In the two top drawings, the trajectory for $t \in [k, k+1]$ is collision-free. However, in the case of the top right drawing the collision checking algorithm will consider the trajectory as non collision-free. In the two bottom drawings, the trajectory for $t \in [k, k+1]$ is not collision-free. In the case of the bottom left drawing, both $\phi_x(k, s, \mathbf{u})$ and $\phi_x(k+1, s, \mathbf{u})$ are collision-free. However, $x(k)$ is not collision-free. This case illustrates that checking whether the endpoints are collision-free is not sufficient, which justifies the use of $x(k)$.

Robots are generated at the origin of each path randomly at a constant rate. Basically, at each time slot, for each path, a random value between 0 and 1 according to a uniform distribution is taken, and if this value exceeds a certain threshold, a robot is generated on this path. The value of this threshold is precisely the generation rate at the path. 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 noticed in [35], 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 see that the robots that are not already accepted in the control area stop way before potential collision configurations).

Finally, the priority assignment policy of Subsection 7.2.1 is simplified based on heuristic considerations. To decide whether robot i can be accepted or not, we need to check whether, under maximum throttle command, it can go through the intersection after all robots already accepted in the control area. To do so, note first that it is sufficient to check if it is the case for the lastly accepted robot of each intersecting path. Now, assume that robot j is the lastly accepted robot on path γ_j. Intuitively, it is clear that, under maximum throttle command, robot i can go through the intersection after robot j, if and only if there is a sufficient time offset between their entries. To this purpose, we compute:

- τ_i : the number of time slots necessary for robot i, under maximum throttle command, to reach "the entry of the collision area between paths γ_i and γ_j", i.e., to reach position $\min\{x_i : x \in \chi_{ij}^{\text{obs}}\}$;
- τ_j : the number of time slots necessary for robot j, under maximum throttle command, to reach "the exit the collision area between paths γ_i and γ_j", i.e., to reach position $\max\{x_j : x \in \chi_{ij}^{\text{obs}}\}$.

Our heuristic approach considers that, under maximum throttle command, robot i can go through the intersection after robot j if and only if $\tau_i \geq \tau_j$. It looks quite natural as it means that robot i should "enter the collision area between paths γ_i and γ_j" after robot j exits this area. Naturally, due, e.g., to the brake safety constraint, this is not equivalent to the formulation of Subsection 7.2.1. However, it is much easier to implement and checking whether the heuristic condition is satisfied is also much less time consuming. Simulation results of Subsection 8.2.1 confirm the efficiency of our heuristic approach as robots seem to enter the control area at the right time, so as to go through the intersection at maximum speed.

8.2 Simulation results

The purpose of the presented simulations is fourfold; they aim to demonstrate:

(a) the ability of priority-based coordination to carry out as complex scheduling as with plan-as-program approaches;

(b) the robustness enabled by planning priorities instead of precise trajectories, making possible to handle unexpected events requiring braking without re-planning, making also possible to deal with bounded, possibly time-varying, uncertainty;

(c) and the ability of priority assignment policies to implement back-pressure algorithms guaranteeing queues stability and opening avenues for the control of a network of autonomous intersections.

8.2.1 Simulations under deterministic control

The experimental intersection is depicted in Figure 8.2. 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. As communication aspects are not considered in this simulation setting, there is no delay for the intersection controller to respond to requests, so robots do not need to anticipate their entry and we take $\delta \equiv 0$, i.e., robots request the entry of the control area if $x_i^{\text{stop}}(s_i) > x_i^{\text{entry}}$, that is just in time.

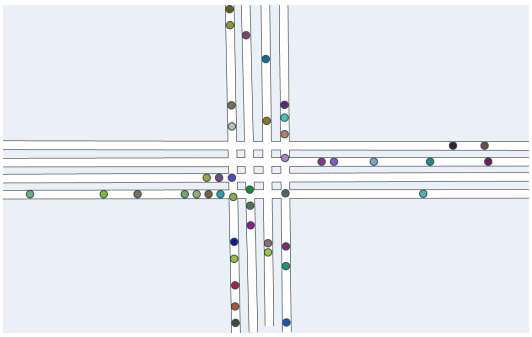


Figure 8.2: The intersection composed of eight straight paths used for simulations.

A video capture of the simulation for an arrival rate of 0.04 robots per time slot on each path is available here[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 0.08 robots per time slot available here[2]. At such an arrival

[1] http://youtu.be/T5ASnKuJLT4
[2] http://youtu.be/tYC6m7Z-S3Y

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, it just appears that priority-based coordination enables to carry out as complex scheduling as traditional approaches using a plan-as-program approach, e.g., [35]. However, the benefit of the priority-based approach is not visible, because in the absence of uncertainty, the control law under assigned priorities always returns $\overline{u}$, it is very similar to an open-loop plan execution.

8.2.2 Robustness regarding unexpected deceleration

Here, to illustrate the robustness of the proposed coordination system with respect to unexpected events requiring deceleration, we consider a scenario in which robots may decide to brake within the control area unexpectedly. The intersection controller, when assigning priorities, does not know that the robot is going to brake within the control area. At the beginning of every time slot, each robot i may switch from a controlled regime under the control law g^G to an unexpected deceleration under constant control $\underline{u}_i$, and vice versa, with probability transitions displayed in Figure 8.3. The probability values p, q are chosen arbitrarily, as the goal is is

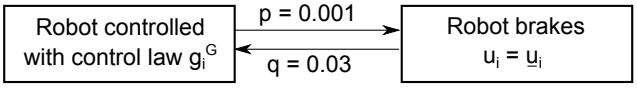


Figure 8.3: Non-deterministic transitions between control regimes

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 modeling 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 robot to slow down unexpectedly.

A video capture of the simulation for an arrival rate of 0.04 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 robots eventually exit the intersection, although the trajectory may be very far from the trajectory under perfect control law.

[3] http://youtu.be/8Xz3S_OhK80

8.2.3 Robustness regarding bounded uncertainty

The simulation results that follow aim at demonstrating the robustness of priority-based coordination in the presence of bounded uncertainty in sensing and control. The same inertia/geometrical parameters as for the previous simulations are used. However, uncertainty is additionally considered.

First of all, we assume the presence of control uncertainty, so the dynamics of robots is described by Equations (6.1)-(6.2). In the presented simulations, the value of control uncertainty bounds are different for each robot. Their average values are (here, n denotes the total number of robots through the simulation run and the sum is over all these robots):

$$\frac{1}{n}\sum_i |\underline{d}_i^v| = \frac{1}{n}\sum_i \overline{d}_i^v = 0.1\ |\underline{u}_i| \tag{8.3}$$

$$\frac{1}{n}\sum_i |\underline{d}_i^u| = \frac{1}{n}\sum_i \overline{d}_i^u = 0.1\ \overline{u}_i = 0.1\ |\underline{u}_i| \tag{8.4}$$

and the actual control uncertainty bounds on each robot vary between 0 and twice the average values according to a uniform distribution. This enables to illustrate that the proposed approach can deal with different control uncertainty bounds on each robot. In average, the uncertainty in control is 10% of the maximum control value as stated by Equations (8.3) and (8.4).

Uncertainty in sensing is also simulated and again, as for control uncertainty, the value of sensing uncertainty bounds are different for each robot. Let δy_i^x and δy_i^v denote the respective maximum absolute errors in position and velocity observations on robot i, their average values are:

$$\frac{1}{n}\sum_i \delta y_i^v = 0.1\ \overline{v}_i \tag{8.5}$$

$$\frac{1}{n}\sum_i \delta y_i^x = D/2 \tag{8.6}$$

and again, the actual observation uncertainty bounds on each robot vary between 0 and twice the average values according to a uniform distribution. In average, the uncertainty in position is one radius of robot and the uncertainty in velocity is 10% of the maximum velocity as stated by Equations (8.5) and (8.6).

Note that to decide to accept or not a robot in the control area, the intersection controller can only access to the non-deterministic state of robots. The heuristic approach is adapted to deal with that and the values of τ_i and τ_j (see Section 8.1), which are necessary to decide to accept or not a robot in the control area, are computed based on the average state of robots considering all possible current true states.

A video capture of the simulation for an arrival rate of 0.02 robots per time slot on each path is available here[4]. The red segments represent the set of positions

[4] http://youtu.be/vpqHbNE6smM

where a robot believes it is located in. One can see that no collision occurs, neither between robots, nor between the red segments. It confirms that the control law in the non-deterministic information space proposed in Chapter 6 results in a collision-free trajectory of the non-deterministic information state.

Finally, to demonstrate the robustness of our approach regarding time-varying uncertainty, we consider a scenario where uncertainty on the observation of position is much higher during a limited time period. In the following simulations, all the parameters are unchanged, but between time slots $t = 500$ and $t = 1000$, the uncertainty on position measures is suddenly multiplied by a factor 10. A video capture of the simulation for an arrival rate of 0.02 robots per time slot on each path is available here[5]. It is remarkable that such a change in the uncertainty of position observation can be handled in a completely reactive manner. Note also that, interestingly, the priority assignment policy during the period of large position uncertainty demonstrates an emerging traffic signal like behavior.

8.2.4 Stability guarantees under back-pressure control

The following simulation results illustrate the ability of priority-based coordination to ensure both efficiency in term of travel time at low traffic density and stability of the queue lengths at high traffic density, in an adaptive manner. The adaptive priority assignment policy proposed in Subsection 7.2.3 has been implemented with the same inertia/geometrical parameters as the simulations presented previously and simulations results are presented in Figure 8.4. For the phase duration, we take $T = 100$ time slots and the threshold is $\Delta Q^{\lim} = 30$. We compare the behavior of the system under the simple priority assignment policy presented in Section 7.2 (referred as "NO BP" in Figure 8.4) versus the adaptive priority assignment policy proposed here (referred as "BP" in Figure 8.4). The top drawings provide the evolution of the arrival rate through time for the two scenarios considered. The drawings in the center depict the evolution of the sum of the four queue lengths ($Q = Q_{11} + Q_{12} + Q_{21} + Q_{22}$) through time. Finally, the bottom drawings represent the average total time spent by robots currently in the intersection. On the left drawings of Figure 8.4, it is clear that at low traffic density, the behavior under both policies are identical: it is not surprising as the queue difference is not likely to be greater than the threshold, so that the two priority assignment policies coincide most of the time. On the right drawings of Figure 8.4, it appears that the queues get unstable under the priority assignment policy of Section 7.2 for an arrival rate of 0.12 robots per time slot. By contrast, under adaptive priority assignment policy, queues are stable. In conclusion, the proposed adaptive priority assignment policy combines the efficiency of the priority assignment policy of Section 7.2 at low traffic density and the stability guarantees of back-pressure control at high traffic density.

[5] http://youtu.be/k14t-fYpy3g

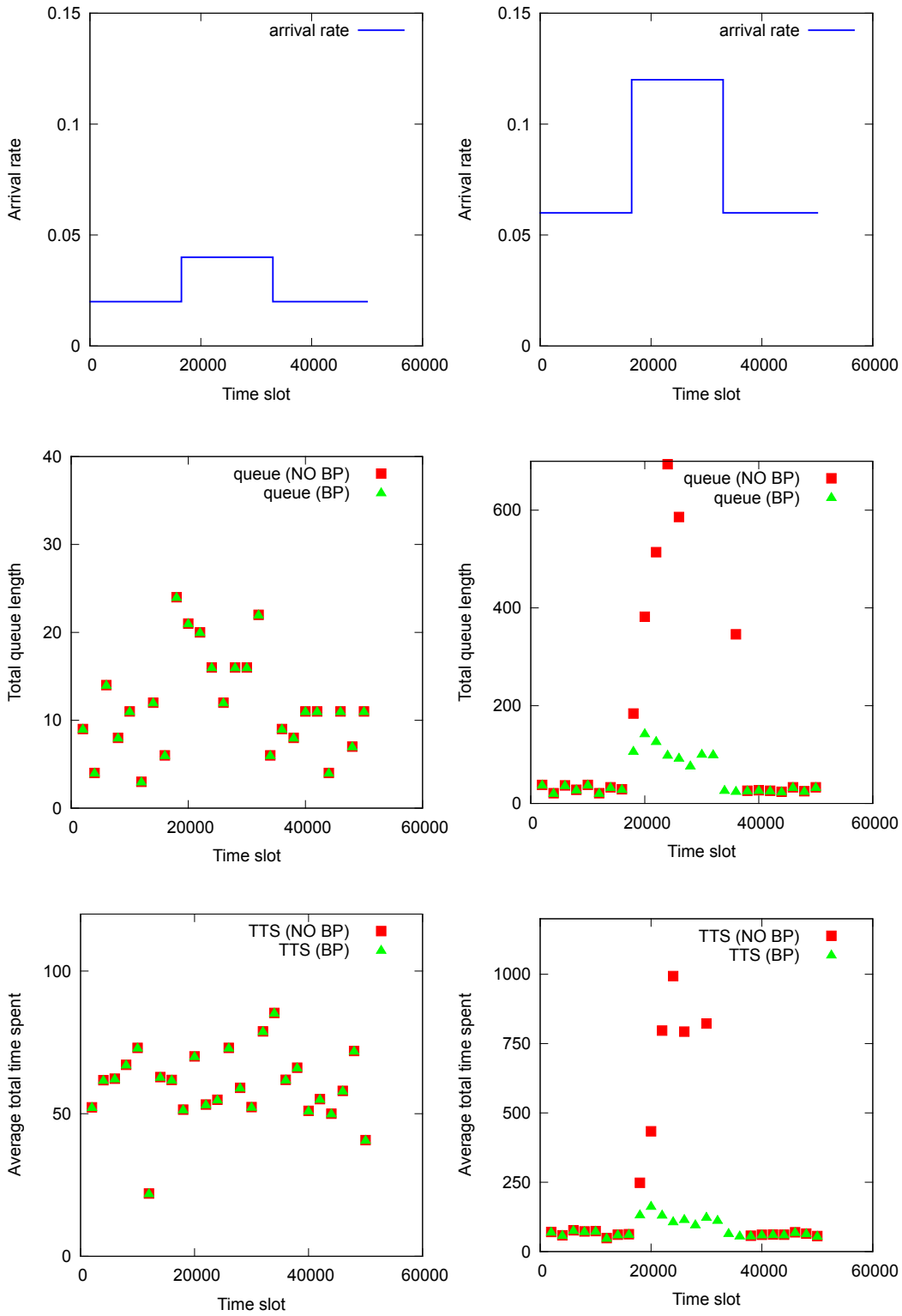


Figure 8.4: Simulations under adaptive priority assignment policy for two arrival rate scenarios.

Conclusions

The present part proposed a priority-based coordination system adopting a three-layer architecture integrating both priority assignment – in the deliberative layer – and priority preserving control – in the behavior-based layer – which are executed in parallel and interface through the sequencing layer. The proposed coordination system is able to manage continuous arrivals of robots at the intersection and to assign priorities dynamically. The proposed architecture takes full benefit of the brake safety property of priority preserving control of Part II as some behavior may require a robot to brake at any moment to handle some unexpected event with the guarantee that priorities will be respected. Priority-based coordination demonstrates both planning and reactive abilities.

In the proposed setting, priority assignment is centralized and is processed by the intersection controller. It is the planning – time consuming – task: assigning efficient priorities requires reasoning about the future. Interestingly, previous work can be used to design efficient priority assignment policies. Priorities are indeed a byproduct of existing trajectory planning algorithms as it suffices to assign the priorities induced by the output planned trajectory. That is why little attention has been paid to the design of efficient priority assignment policies in this part. However, what is quite novel is that, some algorithms commonly used for the control of traffic signals proved useful to assign priorities at high traffic densities as it becomes more efficient to have a traffic-signal-like behavior. Interestingly, priority assignment can adapt to traffic load resulting in a traffic-signal-like behavior ensuring queue lengths stability at high traffic density, and in a much more complex scheduling through combinatorial optimization of priorities to minimize travel delays at low traffic density. This brings elements to the debate about whether autonomous intersection management can really outperform traffic signals. We believe that at high traffic density, a traffic-signal-like behavior is likely to be an optimal strategy as it enables to move robots in platoons, maximizing throughput. However, at low traffic density, optimizing priorities can significantly decrease travel delays and outperform traffic signals. The results of the present part demonstrate that priority-based coordination is able to perform both strategies adaptively.

As highlighted in the introduction of the thesis, a key motivation to the automation of transportation systems is a reduced need of costly infrastructure. Therefore, a major concern of the architecture proposed here is that it requires a central agent at each junction with processing capabilities. However, the only task which is centralized is priority assignment. We believe that the conciseness of the priority graph that merely maps each couple of robot identifiers to a binary value encoding the priority, is a real asset to design a distributed priority assignment policy. However, if first-come-first-serve distributed priority assignment is likely to be quite easily designed, distributed priority assignment performing some combinatorial optimization will be a much more challenging task.

Conclusions and perspectives

Conclusions

This thesis proposed to study the coordination of mobile robots at intersections espousing an approach considering planning as the computation of resources to guide – not control – robots. As a result, the plan representation appeared to differ widely from what it is in traditional motion planning. In traditional approaches, planning consists in computing a planned reference trajectory that robots must execute. The reference trajectory constitutes the plan and in the control phase, a control law configured by the reference trajectory is in charge of tracking the planned trajectory. In the approach of this thesis, the plan is the priority graph encoding the particular homotopy class chosen to solve the coordination problem and to our knowledge, there existed no standard control scheme to ensure the described trajectory belongs to the chosen homotopy class, i.e., to ensure priorities are respected. In priority-based coordination, as there is no reference trajectory, there is no trajectory tracking which is replaced by so-called priority preserving control. Our approach using priorities as a plan to guide – not control – robots confirmed suitable to coordinate multiple robots at an intersection area, endowing the system with robustness properties. Part I provides a powerful tool to characterize the structure of solutions to the coordination problem: the priority graph. Previous work already noticed the existence of homotopy classes of feasible paths in the coordination space, yet without providing a meaningful representative of homotopy classes. The main contribution of the first part of the thesis is to provide such a meaningful representative: the priority graph. Priorities uniquely encode the homotopy classes of feasible paths. Choosing a particular priority graph to coordinate robots appears as the discrete part of the coordination problem. It thus provides a geometrical understanding in the coordination space of why planning priorities instead of a precise trajectory results in an increased robustness. It merely appears as the consequence of constraining the path of robots in the coordination space to remain in a homotopy class – a large set of feasible paths continuously deformable into each other –, instead of assigning a particular precise feasible path to follow. The ”size” of the homotopy class provides some freedom of action. Part II demonstrates that robots can easily go through the intersection while respecting priorities in a reactive manner. Under assigned priorities, the combinatorial complexity of multi robot control is avoided as for each pair of robots there is not two strategies to avoid collisions anymore: the robot with lower priority must decelerate in favor of the robot with higher priority. It is thus not surprising that priority preserving control can be carried out in polynomial time. From the coordination space point of view, priority preserving control ensures that the trajectory described by the multi robot system belongs to the homotopy class encoded by the assigned priorities. To this purpose, a control law is configured by the assigned priorities and is in charge of priority preservation. It is very different from the trajectory tracking approach as it allows for example all robots to stop for a while to handle some unexpected event while respecting priorities, i.e., without replanning. By contrast, in traditional motion planning, the plan must be executed, and the only

way to recover some freedom of action is replanning. In the absence of inertia, the control law proposed in Chapter 4 ensures that the resulting trajectory is optimal for the assigned priorities, recovering the existence of a local optimum in each homotopy class [47]. Even though the dynamics model used in Chapter 5 is quite simple, it convinces that the additional complexity in the presence of kinodynamic constraints can be easily tackled by introducing the brake safety constraint. The byproduct of the conservative brake safety constraint is an increased robustness regarding unexpected deceleration of robots. Moreover, the proposed control scheme is decentralized as the output of the control law can be computed on each robot independently, thus not requiring any form of agreement through communication links. This valuable benefit is allowed by the prior agreement on the priority graph which is done at the planning level. This prior agreement which requires some form of communication enables to select a particular strategy for collision avoidance – the priority graph –, so that no more agreement is required at the control level which can be decentralized. Finally, Chapter 6 provides some elements to control the system in the presence of uncertainty. It shows that bounded uncertainty can be handled by considering worst-case scenarios. Even though the uncertainty model is quite simple, it demonstrates that the key benefit of priority-based coordination is its ability to handle uncertainty in a reactive manner. For example, robots may stop for a while if uncertainty is very large due to communication concerns and restart without replanning, merely using a control law in the information space. Part III has a more engineering flavor and proposes a three-layer architecture integrating both priority assignment and priority preserving control which are executed in parallel. Priority assignment is carried out by a central deliberative intersection controller. Robots implement multiple behaviors including one ensuring priorities are respected. Robot's sequencer interfaces the reactive and the deliberative layers through asynchronous communication with the intersection controller to negotiate the entry of the control area and by activating/deactivating/configuring primitive behaviors. Compared to traditional plan-as-program approaches, robots retain reactive capabilities through the intersection. A large class of unexpected events – all events requiring braking – can be handled in a reactive manner without need to replan endowing the system with significant robustness. This thesis proposes a novel class of coordination systems at intersections – using priorities to guide robots – and therefore still suffers from some limitations and opens several perspectives for future work.

Limitations and perspectives

From the theoretical point of view

Homotopy classes under imperfect lateral control The path-following assumption of Figure 2.1 is key to the definition of priorities and to the existence of homotopy classes of feasible paths uniquely encoded by priorities. In real systems, perfect path following cannot be guaranteed as lateral control is based on

imperfect mapping/localization data and imperfect actuators. Hence, future work should investigate which assumptions on lateral control still guarantee all the results of Part I which is the foundation of the priority-based approach. We believe that under bounded uncertainty on lateral control, the results of Part I can be extended by considering the worst-case obstacle region considering all possible geometric paths. However, this could decrease performance and it would raise the problem of handling a lateral error beyond the fixed bound.

Dealing with partial information In the current setting, all robots are assumed to know the assigned priorities and the current state of other robots. Even though Chapter 6 provides elements on how to deal with bounded uncertainty, it still assumes that all robots know the complete priority graph and the current non-deterministic information state of all other robots, which is still a strong assumption. Further work should focus on relaxing these assumptions. We believe that priority-based coordination has real strengths to deal with partial information concerns. First of all, some priorities are redundant, as if robots 2 and 3 travel along the same path (say 3 follows 2) and have both priority over robot 1, then robot 1 only needs to know that it has priority over the first of the two robots, i.e., robot 2. Moreover, to execute the priority preserving control law, a given robot only needs to know the current state of robots in the neighborhood as there is no need to anticipate beyond a certain area.

Distributed priority assignment As highlighted in the conclusion of Part III, while a key motivation to the automation of transportation systems is a reduced need of costly infrastructure, the proposed three-layer architecture requires a central agent at each junction with processing capabilities. Distributed priority assignment would imply a consensus algorithm as all robots need to agree on a common decision: the assigned priorities. We believe that without efficiency considerations, previous work on consensus algorithms should help to design simple distributed priority assignment policies, e.g., a first-come-first-serve policy. However, optimizing priorities requires to perform time consuming algorithms reasoning about the future and a distributed implementation of such algorithms should prove challenging.

From the application point of view

Challenges for an implementation in real systems First of all, localization and mapping aspects have not been addressed in this thesis and are challenges in themselves. These topics are intensive research fields, both in robotics and intelligent transportation systems communities. Priority-based coordination requires building a map specifying predefined geometric paths to go through the intersection and robots need to have an estimate of their position on their path. We believe that for an application in self-driven vehicles, existing maps of the road network and lane markings/panels – more generally, the physical infrastructure – should help localization and mapping tasks. Communication is another important aspect that has been

left behind. Standardized messages should be designed to support priority-based coordination, taking into account constraints in terms of delay and amount of data. This is one of the tasks currently achieved in the European project Autonet2030 where a complete cooperative system architecture is designed for the cooperation of intelligent vehicles supporting, in particular, coordination at intersections. We believe that the conciseness of the plan representation – the priority graph is merely mapping couple of robot identifiers to a binary value – is a valuable feature of priority-based coordination as it limits the amount of data to be exchanged. By contrast, plan-as-program approaches need to exchange precise trajectories which include much more data.

Sharing the road between autonomous and human-driven vehicles Autonomous vehicles will arrive gradually, and they will have to "share the road" for a while. According to [27], only 50% of vehicles will be autonomous by 2030. Cooperation between autonomous/semi-autonomous/human-driven vehicles thus appears necessary. As respecting priorities is a capability of both humans and robots, the priority-based approach is particularly adapted for the development of algorithms aiming at coordinate both human-driven and autonomous vehicles. In [108], a priority-based autonomous intersection management system is proposed in this context of "mixed traffic flow". Priorities are assigned by an intersection controller, yet human-driven vehicles are not aware of that, and just respect traffic signals as in a usual signalized intersection. A video capture of simulations is available here[6].

Energy efficient priority preserving control Priority preserving control proposed in Part II is a kind of bang-bang control switching quite abruptly between maximum brake and maximum throttle. Such control law, while priority preserving, raises energy efficiency concerns as abrupt control switches are energy consuming. To this purpose, we believe that model predictive control with a cost function accounting for both priority preservation, delay reduction and energy efficiency should prove useful by anticipating the need to brake and result in smoother trajectories as demonstrated by recent work [88].

Handling priority violation The present work assumed robots respect the assigned priorities. We are convinced that the ability to respect priorities is a must that self-driven vehicles need to possess to be deployed on the roads. People would accept that vehicles cannot follow a precise trajectory through an intersection. By contrast, they would not accept an autonomous vehicle unable to respect assigned priorities. That is why we believe that priority violations should occur mainly under major system failure of one robot. Another reason why it can occur in an application to self-driven cars is when a driver decides to stop the self-driving system and to take back the control of the vehicle for some unexpected reason. Such circumstances would require both priority violation detection and real-time dynamic

[6] http://www.youtube.com/watch?v=L3B_FrNn_Pk

priority assignment with all the attention paid on safety, i.e., collision avoidance.

Towards a network of autonomous intersections All the work around traffic signal control demonstrates that controlling a network of intersections is of high complexity. Recent work based on back-pressure algorithms tend do demonstrate that queues stability guarantees of the network can be obtained, while each intersection controller uses only local information. Future work should extend the simple back-pressure priority assignment policy at a single intersection proposed in Chapter 7 and consider adaptive priority assignment at a network of autonomous intersections using back-pressure algorithms.

Towards coordination of aerial drones Finally, even though this work was originally motivated by applications in autonomous vehicles on roadways, applications to other fields should be investigated. For instance, the results of this thesis may be applicable to coordinate aerial drones in a three-dimensional space provided the geometric three-dimensional paths followed by drones are fixed. If so, each drone i still has only one degree of freedom – its curvilinear coordinate along its three-dimensionnal path $\gamma_i \subset \mathbb{R}^3$ – and the priority-based framework proposed in this thesis is still applicable.

Bibliography

[1] Cybercars. http://www.cybercars.org/, 2006. Accessed: 2014-06-11. (Cited on page 4.)

[2] Citymobil - towards advanced road transport for the urban environment. http://http://www.citymobil-project.eu/, 2011. Accessed: 2014-06-11. (Cited on page 1.)

[3] Citymobil2 - cities demonstrating automated road passenger transport. http://http://http://www.citymobil2.eu//, 2012. Accessed: 2014-06-11. (Cited on page 1.)

[4] Autonet2030 (2014): Co-operative systems in support of networked automated driving by 2030. https://www.autonet2030.eu, 2014. Accessed: 2014-06-11. (Cited on page 2.)

[5] K Aboudolas, M Papageorgiou, and E Kosmatopoulos. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 17(2):163–174, 2009. (Cited on page 11.)

[6] Philip E Agre and David Chapman. What are plans for? *Robotics and autonomous systems*, 6(1):17–34, 1990. (Cited on pages 2, 3, 8 and 9.)

[7] S. Akella and S. Hutchinson. Coordinating the motions of multiple robots with specified trajectories. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 1, pages 624 – 631 vol.1, 2002. (Cited on pages 6 and 105.)

[8] Luis Alvarez and Roberto Horowitz. Safe platooning in automated highway systems. Institute of Transportation Studies, Research Reports, Working Papers, Proceedings qt1v97t5w1, Institute of Transportation Studies, UC Berkeley, January 1997. (Cited on page 1.)

[9] David Angeli and Eduardo D Sontag. Monotone control systems. *Automatic Control, IEEE Transactions on*, 48(10):1684–1698, 2003. (Cited on pages 63, 64 and 72.)

[10] T. Araki, Y. Sugiyama, and T. Kasami. Complexity of the deadlock avoidance problem. 1977. (Cited on page 8.)

[11] Tsz-Chiu Au, Neda Shahidi, and Peter Stone. Enforcing liveness in autonomous traffic management. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence*, pages 1317–1322, August 2011. (Cited on pages 4 and 107.)

[12] Jerome Barraquand and Jean-Claude Latombe. Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, 10(6):628–649, 1991. (Cited on page 4.)

[13] Kostas E Bekris, Konstantinos I Tsianos, and Lydia E Kavraki. A distributed protocol for safe real-time planning of communicating vehicles with second-order dynamics. In *Proceedings of the 1st international conference on Robot communication and coordination*, page 9. IEEE Press, 2007. (Cited on pages 6 and 73.)

[14] Kostas E Bekris, Konstantinos I Tsianos, and Lydia E Kavraki. Safe and distributed kinodynamic replanning for vehicular networks. *Mobile Networks and Applications*, 14(3):292–308, 2009. (Cited on page 6.)

[15] Rodrigo Benenson, Stéphane Petti, Thierry Fraichard, and Michel Parent, Null. Towards urban driverless vehicles. *The International Journal of Robotics Research*, 1/2(6):4 – 23, 2008. (Cited on page 4.)

[16] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. Optimizing schedules for prioritized path planning of multi-robot systems. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 271–276. IEEE, 2001. (Cited on page 6.)

[17] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 41(2):89–99, 2002. (Cited on page 6.)

[18] Zeungnam Bien and Jihong Lee. A minimum-time trajectory planning method for two robots. *Robotics and Automation, IEEE Transactions on*, 8(3):414–418, 1992. (Cited on page 5.)

[19] Joan S Birman. *Braids, links, and mapping class groups*. Number 82. Princeton University Press, 1974. (Cited on page 39.)

[20] Sara Bouraine, Thierry Fraichard, and Hassen Salhi. Relaxing the inevitable collision state concept to address provably safe mobile robot navigation with limited field-of-views in unknown dynamic environments. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2985 –2991, sept. 2011. (Cited on page 73.)

[21] Rodney A Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14–23, 1986. (Cited on pages 2, 6 and 103.)

[22] Cheol Chang, Myung Jin Chung, and Bum Hee Lee. Collision avoidance of two general robot manipulators by minimum delay time. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(3):517–522, 1994. (Cited on page 5.)

[23] Christopher M Clark, Tim Bretl, and Stephen Rock. Applying kinodynamic randomized motion planning with a dynamic priority system to multi-robot space systems. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 7, pages 7–3621. IEEE, 2002. (Cited on page 6.)

[24] E. G. Coffman and M. J. Elphick. System deadlocks. *Computing Surveys*, 3:67–78, 1971. (Cited on pages 7, 9 and 45.)

[25] A. Colombo and D. Del Vecchio. Efficient algorithms for collision avoidance at intersections. *Hybrid Systems: Computation and Control*, 2012. (Cited on pages 13, 59, 71, 73 and 93.)

[26] Francisco de Asis Garcia Collado. Développement d'un algorithme pour le passage des carrefours. Master's thesis, Mines ParisTech, 2010. (Cited on page 4.)

[27] Arnaud de La Fortelle, Xiangjun Qian, Sébastien Diemer, Jean Grégoire, Fabien Moutarde, Silvère Bonnabel, Ali Marjovi, Ignacio Llatser Andreas Festag, and Sjöberg Katrin. Network of automated vehicles: The autonet2030 vision. In *ITS World Congress*, 2014. (Cited on page 124.)

[28] D. Del Vecchio, M. Malisoff, and R. Verma. A separation principle for a class of hybrid automata on a partial order. In *American Control Conference, 2009. ACC '09.*, pages 3638 –3643, june 2009. (Cited on pages 71 and 73.)

[29] A. Di Febbraro, D. Giglio, and N. Sacco. On applying petri nets to determine optimal offsets for coordinated traffic light timings. In *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, pages 773–778, 2002. (Cited on page 11.)

[30] Christina Diakaki, Markos Papageorgiou, and Kostas Aboudolas. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, 10(2):183–195, 2002. (Cited on page 11.)

[31] Dimos V Dimarogonas, Kostas J Kyriakopoulos, and Dimitris Theodorakatos. Totally distributed motion control of sphere world multi-agent systems using decentralized navigation functions. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2430–2435. IEEE, 2006. (Cited on page 10.)

[32] Dimos V Dimarogonas, Savvas G Loizou, Kostas J Kyriakopoulos, and Michael M Zavlanos. Decentralized feedback stabilization and collision avoidance of multiple agents. *NTUA, http://users. ntua. gr/ddimar/TechRep0401. pdf, Tech. Report*, 2004. (Cited on page 10.)

[33] K. Dresner and P. Stone. Multiagent traffic management: a reservation-based intersection control mechanism. In *Autonomous Agents and Multiagent*

Systems, 2004. AAMAS 2004. Proceedings of the Third International Joint Conference on, pages 530 –537, july 2004. (Cited on pages 3 and 12.)

[34] Kurt Dresner and Peter Stone. Mitigating catastrophic failure at intersections of autonomous vehicles. In *AAMAS Workshop on Agents in Traffic and Transportation*, pages 78–85, Estoril, Portugal, May 2008. (Cited on page 4.)

[35] Kurt Dresner and Peter Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, March 2008. (Cited on pages 3, 6, 105, 107, 111 and 113.)

[36] Chris Elsaesser and Marc G Slack. Integrating deliberative planning in a robot architecture. In *NASA CONFERENCE PUBLICATION*, pages 782–782. NASA, 1994. (Cited on page 2.)

[37] Michael Erdmann and Tomas Lozano-Perez. On multiple moving objects. *Algorithmica*, 2(1-4):477–521, 1987. (Cited on page 6.)

[38] Aria Etemad. interactive – accident avoidance by active intervention for intelligent vehicles. In *20th ITS World Congress Tokyo 2013*, 2013. (Cited on page 1.)

[39] Alessandro Farinelli, Luca Iocchi, and Daniele Nardi. Multirobot systems: a classification focused on coordination. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(5):2015–2028, 2004. (Cited on page 2.)

[40] T. Fraichard and 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*, pages 466 –472, sep 1989. (Cited on page 5.)

[41] Thierry Fraichard and Hajime Asama. Inevitable collision states - a step towards safer robots? *Advanced Robotics -Utrecht-*, 18(10):1001–1024, 2004. (Cited on page 73.)

[42] Nathan H Gartner. Opac: A demand-responsive strategy for traffic signal control. *Transportation Research Record*, (906), 1983. (Cited on page 11.)

[43] Nathan H Gartner, John DC Little, and Henry Gabbay. Optimization of traffic signal settings by mixed-integer linear programming part i: The network coordination problem. *Transportation Science*, 9(4):321–343, 1975. (Cited on page 11.)

[44] Erann Gat. Three-layer architectures. In *Artificial intelligence and mobile robots*, pages 195–210. MIT Press, 1998. (Cited on pages 97, 101 and 104.)

[45] R. Ghaemi and D. Del Vecchio. Control for safety specifications of systems with imperfect information on a partial order. *Automatic Control, IEEE Transactions on*, 59(4):982–995, April 2014. (Cited on page 84.)

[46] Robert Ghrist and Steven M. Lavalle. Nonpositive curvature and pareto optimal coordination of robots. *SIAM J. Control Optim.*, 45:1697–1713, November 2006. (Cited on pages 5 and 68.)

[47] Robert Ghrist, Jason M O'Kane, and Steven M LaValle. Computing pareto optimal coordinations on roadmaps. *The International Journal of Robotics Research*, 12:997–1012, 2005. (Cited on pages 5, 17, 39, 55, 68, 93 and 122.)

[48] Jean Gregoire, Silvère Bonnabel, and Arnaud 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. (Cited on page 37.)

[49] Jean Gregoire, Silvère Bonnabel, and Arnaud de La Fortelle. Priority-based coordination of robots. *arXiv preprint arXiv:1306.0785*, 2013. (Cited on pages 71 and 97.)

[50] Jean Gregoire, Silvere Bonnabel, and Arnaud de La Fortelle. Priority-based intersection management with kinodynamic constraints. In *Proceedings of the European Control Conference, ECC'14*, 2014. (Cited on page 71.)

[51] Jean Gregoire, Emilio Frazzoli, Arnaud de La Fortelle, and Tichakorn Wongpiromsarn. Back-pressure traffic signal control with unknown routing rates. In *IFAC World Congress, Cape Town, South Africa*, 2014. (Cited on pages 12 and 107.)

[52] Jean Gregoire, Emilio Frazzoli, Arnaud de La Fortelle, and Tichakorn Wongpiromsarn. Capacity-aware back-pressure traffic signal control. *IEEE Transactions on Control of Network Systems*, 2014. Accepted. (Cited on pages 12 and 107.)

[53] NM Gregoire, A Gjedde, F Plum, and TE Duffy. Cerebral blood flow and cerebral metabolic rates for oxygen, glucose, and ketone bodies in newborn dogs. *Journal of neurochemistry*, 30(1):63–69, 1978. (Cited on page iv.)

[54] Yi Guo and L.E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3, pages 2612 –2619, 2002. (Cited on page 5.)

[55] MR Hafner, D Cunningham, L Caminiti, and D Del Vecchio. Automated vehicle-to-vehicle collision avoidance at intersections. In *Proceedings of World Congress on Intelligent Transport Systems*, 2011. (Cited on page 71.)

[56] Matthew Hausknecht, Tsz-Chiu Au, and Peter Stone. Autonomous intersection management: Multi-intersection optimization. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2011. (Cited on page 4.)

[57] Jean-Jacques Henry, Jean-Loup Farges, and J Tuffal. The prodyn real time traffic algorithm. In *Proceedings of the 4th IFAC/IFORS Conference on Control in Transportation Systems,*, 1984. (Cited on page 11.)

[58] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects: *pspace*-hardness of the 'warehouseman's problem'. *The International Journal of Robotics Research*, 3(4):76–88, 1984. (Cited on page 4.)

[59] R. Horowitz and P. Varaiya. Control design of an automated highway system. *Proceedings of the IEEE*, 88(7):913 –925, jul 2000. (Cited on page 1.)

[60] Jianghai Hu, Maria Prandini, and Shankar Sastry. Optimal coordinated motions of multiple agents moving on a plane. *SIAM Journal on Control and Optimization*, 42(2):637–668, 2003. (Cited on page 39.)

[61] Sze-tsen Hu and Valerie Hu. *Homotopy theory*, volume 8. Academic press, 1959. (Cited on page 39.)

[62] PB Hunt, DI Robertson, RD Bretherton, and MC Royle. The scoot on-line traffic signal optimisation technique. *Traffic Engineering & Control*, 23(4), 1982. (Cited on page 11.)

[63] Zhong-Ping Jiang and Henk Nijmeijer. Tracking control of mobile robots: a case study in backstepping. *Automatica*, 33(7):1393–1399, 1997. (Cited on pages 55 and 83.)

[64] Arnaud de La Fortelle Jingman Fan and Silvère Bonnabelle. Modélisation et résolution du problème de passages de véhicules dans un carrefour. Technical report, Mines ParisTech – Centre de robotique, 2011. (Cited on page 4.)

[65] K. Kant and S. W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *International Journal of Robotics Research*, 5(3):72–89, 1986. (Cited on page 4.)

[66] S. Karaman and E. Frazzoli. Incremental sampling-based algorithms for optimal motion planning. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010. (Cited on page 4.)

[67] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996. (Cited on page 4.)

[68] Eric C Kerrigan. *Robust constraint satisfaction: Invariant sets and predictive control*. PhD thesis, PhD thesis, Cambridge, 2000. (Cited on pages 64, 65, 75 and 76.)

[69] Jinsuck Kim, Roger A Pearce, and Nancy M Amato. Extracting optimal paths from roadmaps for motion planning. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 2424–2429. IEEE, 2003. (Cited on page 4.)

[70] Julian Kolodko and Ljubo Vlacic. Cooperative autonomous driving at the intelligent control systems laboratory. *Intelligent Systems, IEEE*, 18(4):8–11, 2003. (Cited on page 4.)

[71] H. Kowshik, D. Caveney, and P.R. Kumar. Provable systemwide safety in intelligent intersections. *Vehicular Technology, IEEE Transactions on*, 60(3):804 –818, march 2011. (Cited on pages 12 and 71.)

[72] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA, 1991. (Cited on page 5.)

[73] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/. (Cited on pages 5, 17, 21 and 22.)

[74] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998. (Cited on page 4.)

[75] Steven M LaValle and Seth 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*, volume 4, pages 3034–3039. IEEE, 1996. (Cited on pages 83 and 84.)

[76] Steven M LaValle and Seth A Hutchinson. Optimal motion planning for multiple robots having independent goals. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 3, pages 2847 –2852 vol.3, apr 1996. (Cited on page 5.)

[77] Steven M LaValle and Seth A Hutchinson. An objective-based framework for motion planning under sensing and control uncertainties. *The International Journal of Robotics Research*, 17(1):19–42, 1998. (Cited on pages 83 and 84.)

[78] Mark Lawley and Spyros Reveliotis. Deadlock avoidance for sequential resource allocation systems: Hard and easy cases. *IEEE Transactions on Automatic Control*, 46:1572–1583, 2001. (Cited on page 8.)

[79] Tung Le, Peter Kovacs, Neil Walton, Hai L Vu, Lachlan L Andrew, and Serge S Hoogendoorn. Decentralized signal control for urban road networks. *arXiv preprint arXiv:1310.0491*, 2013. (Cited on page 11.)

[80] Ti-Chung Lee, Kai-Tai Song, Ching-Hung Lee, and Ching-Cheng Teng. Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. *Control Systems Technology, IEEE Transactions on*, 9(2):305–318, 2001. (Cited on pages 55 and 83.)

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

[82] Hong K Lo, Elbert Chang, and Yiu Cho Chan. Dynamic network traffic control. *Transportation Research Part A: Policy and Practice*, 35(8):721–744, 2001. (Cited on page 11.)

[83] Savvas G Loizou and Kostas J Kyriakopoulos. Closed loop navigation for multiple holonomic vehicles. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2861–2866. IEEE, 2002. (Cited on page 10.)

[84] PR Lowrie. Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic. Technical report, 1990. (Cited on page 11.)

[85] Tomas Lozano-Perez. Spatial planning: A configuration space approach. Technical report, 1980. (Cited on pages 4 and 5.)

[86] Vladimir J. Lumelsky and KR Harinarayan. Decentralized motion planning for multiple mobile robots: The cocktail party model. In *Robot colonies*, pages 121–135. Springer, 1997. (Cited on page 7.)

[87] Vladimir J Lumelsky and Alexander A Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2(1-4):403–430, 1987. (Cited on page 65.)

[88] Laleh Makarem and Denis Gillet. Model predictive coordination of autonomous vehicles crossing intersections. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 1799–1804. IEEE, 2013. (Cited on page 124.)

[89] Maja J Mataric. Behavior-based robotics. *MIT Encyclopedia of Cognitive Sciences*, pages 74–77, 1999. (Cited on page 7.)

[90] Adelardo AD Medeiros. A survey of control architectures for autonomous mobile robots. *Journal of the Brazilian Computer Society*, 4(3), 1998. (Cited on page 3.)

[91] Olivier Mehani and Arnaud 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, pages 1159–1166, Berlin, Heidelberg, 2007. Springer-Verlag. (Cited on pages 4 and 6.)

[92] Alain Micaelli, Claude Samson, et al. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. 1993. (Cited on pages 55 and 83.)

[93] Alan J Miller. Settings for fixed-cycle traffic signals. *Operations Research*, pages 373–386, 1963. (Cited on page 11.)

[94] Pitu Mirchandani and Larry Head. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415–432, 2001. (Cited on page 11.)

[95] NCSA. National center for statistics and analysis, traffic safety facts 2003. Technical report, U.S. DOT, Washington, DC, 2004. (Cited on page 2.)

[96] Michael J Neely, Eytan Modiano, and Charles E Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *Selected Areas in Communications, IEEE Journal on*, 23(1):89–103, 2005. (Cited on page 11.)

[97] P.A. O'Donnell and T. Lozano-Periz. Deadlock-free and collision-free coordination of two robot manipulators. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 484 –489 vol.1, may 1989. (Cited on pages 17, 21, 30 and 32.)

[98] Carolina Osorio and Michel Bierlaire. An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *European Journal of Operational Research*, 196(3):996–1007, 2009. (Cited on page 11.)

[99] Carolina Osorio and Michel Bierlaire. A surrogate model for traffic optimization of congested networks: an analytic queueing network approach. *Report TRANSP-OR*, 90825:1–23, 2009. (Cited on page 11.)

[100] Lucia Pallottino, Vincenzo Giovanni Scordio, Antonio Bicchi, and Emilio Frazzoli. Decentralized cooperative policy for conflict resolution in multivehicle systems. *Robotics, IEEE Transactions on*, 23(6):1170–1183, 2007. (Cited on page 7.)

[101] Markos Papageorgiou, Christina Diakaki, Vaya Dinopoulou, Apostolos Kotsialos, and Yibing Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003. (Cited on pages 10 and 11.)

[102] David W Payton. Internalized plans: A representation for action resources. *Robotics and Autonomous Systems*, 6(1):89–103, 1990. (Cited on pages 9 and 10.)

[103] Jufeng Peng and Srinivas Akella. Coordinating the motions of multiple robots with kinodynamic constraints. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 4066–4073. IEEE, 2003. (Cited on page 6.)

[104] Jufeng Peng and Srinivas Akella. Coordinating multiple robots with kinodynamic constraints along specified paths. *The International Journal of Robotics Research*, 24(4):295–310, 2005. (Cited on page 6.)

[105] Guilherme AS Pereira, Aveek K Das, R Vijay Kumar, and Mario FM Campos. Decentralized motion planning for multiple robots subject to sensing and communication constraints. 2003. (Cited on page 10.)

[106] Stephane Petti and Thierry Fraichard. Reactive Planning Under Uncertainty Among Moving Obstacles. In *Proc. of the Int. Symp. on Robotics*, Tokyo (JP), France, November 2005. voir basilic : http://emotion.inrialpes.fr/bibemotion/2005/PF05c/ address: Tokyo (JP). (Cited on page 84.)

[107] Stéphane Petti and Thierry Fraichard. Partial motion planning framework for reactive planning within dynamic environments. In *in AAAI Intl. Conf. ICAR*, 2005. (Cited on page 4.)

[108] Xiangjun Qian, Jean Gregoire, Fabien Moutarde, and Arnaud de La Fortelle. Autonomous intersection management for mixed traffic flow. 2013. (Cited on page 124.)

[109] Ralf Regele and Paul Levi. Cooperative multi-robot path planning by heuristic priority adjustment. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5954–5959. IEEE, 2006. (Cited on page 6.)

[110] S.A. Reveliotis and E. Roszkowska. On the complexity of maximally permissive deadlock avoidance in multi-vehicle traffic systems. *Automatic Control, IEEE Transactions on*, 55(7):1646–1651, 2010. (Cited on page 8.)

[111] Spyros A Reveliotis and Elzbieta Roszkowska. Conflict resolution in free-ranging multivehicle systems: A resource allocation paradigm. *Robotics, IEEE Transactions on*, 27(2):283–296, 2011. (Cited on page 8.)

[112] Elon Rimon and Daniel E Koditschek. Exact robot navigation using artificial potential functions. *Robotics and Automation, IEEE Transactions on*, 8(5):501–518, 1992. (Cited on page 10.)

[113] Elzbieta Roszkowska and Spyros A Reveliotis. On the liveness of guidepath-based, zone-controlled dynamically routed, closed traffic systems. *Automatic Control, IEEE Transactions on*, 53(7):1689–1695, 2008. (Cited on page 8.)

[114] Mitul Saha and Pekka Isto. Multi-robot motion planning by incremental coordination. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5960–5963. IEEE, 2006. (Cited on page 5.)

[115] Jacob T Schwartz and Micha Sharir. On the piano movers' problem: Iii. coordinating the motion of several independent bodies: the special case of circular bodies moving amidst polygonal barriers. *The International Journal of Robotics Research*, 2(3):46–75, 1983. (Cited on pages 4, 7 and 8.)

[116] Jean Serra. *Image analysis and mathematical morphology*. Academic Press, Inc., 1983. (Cited on page 47.)

[117] Vikrant Sharma, Michael A Savchenko, Emilio Frazzoli, and Petros G Voulgaris. Time complexity of sensor-based vehicle routing. In *Robotics: Science and Systems*, pages 297–304. Citeseer, 2005. (Cited on page 5.)

[118] Weihua Sheng, Qingyan Yang, and Yi Guo. Cooperative driving based on inter-vehicle communications: Experimental platform and algorithm. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5073–5078, 2006. (Cited on page 4.)

[119] SP Shepherd. A review of traffic signal control. Technical report, 1992. (Cited on page 11.)

[120] Kang G Shin and Zin Zheng. Minimum-time collision-free trajectory planning for dual-robot systems. *Robotics and Automation, IEEE Transactions on*, 8(5):641–644, 1992. (Cited on page 5.)

[121] S.E. Shladover. Ahs research at the california path program and future ahs research needs. In *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pages 4–5, Sept 2008. (Cited on page 1.)

[122] D Soetanto, L Lapierre, and A Pascoal. Adaptive, non-singular path-following control of dynamic wheeled robots. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 2, pages 1765–1770. IEEE, 2003. (Cited on pages 55 and 83.)

[123] Lucy Suchman. Plans and situated actions. *New York, Cambridge University*, 1986. (Cited on pages 3 and 6.)

[124] S-H Suh and Kang G Shin. A variational dynamic programming approach to robot-path planning with a distance-safety criterion. *Robotics and Automation, IEEE Journal of*, 4(3):334–349, 1988. (Cited on page 4.)

[125] Richard S Sutton. Planning by incremental dynamic programming. In *ML*, pages 353–357. Citeseer, 1991. (Cited on page 4.)

[126] P. Svestka and M.H. Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 2, pages 1631–1636 vol.2, 1995. (Cited on page 4.)

[127] Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, 1992. (Cited on pages 11 and 107.)

[128] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006. (Cited on page 1.)

[129] Emanuel Todorov. *Optimal control theory*. MIT Press, Cambridge, 2006. (Cited on pages 55 and 83.)

[130] J.R. Treat, N.J. Castellan, R.L. Stansifer, R.E. Mayer, R.D. Hume, D. Shinar, S.T. McDonald, and N.S. Tumbas. *Tri-level Study of the Causes of Traffic Accidents: Final Report. Volume I: Causal Factor Tabulations and Assessments*. 1977. (Cited on page 2.)

[131] Jur Van Den Berg, Pieter Abbeel, and Ken Goldberg. Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7):895–913, 2011. (Cited on page 83.)

[132] Jur P Van Den Berg and Mark H Overmars. Prioritized motion planning for multiple robots. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 430–435. IEEE, 2005. (Cited on pages 6 and 12.)

[133] Jan P van Dijke. Citymobil, advanced road transport for the urban environment. final results. In *18th ITS World Congress*, 2011. (Cited on page 1.)

[134] Pravin Varaiya. A universal feedback control policy for arbitrary networks of signalized intersections. Technical report, 2009. (Cited on pages 11, 12 and 107.)

[135] Pravin Varaiya. The max-pressure controller for arbitrary networks of signalized intersections. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*, pages 27–66. Springer, 2013. (Cited on page 11.)

[136] Rajeev Verma and Domitilla Del Vecchio. Safety control of hidden mode hybrid systems. *Automatic Control, IEEE Transactions on*, 57(1):62–77, 2012. (Cited on page 71.)

[137] Richard Wallace and G Silberg. Self-driving cars: the next revolution. *KPMG and CAR (Center for Automotive research), source: http://www. kpmg. com/Ca/en/IssuesAndInsights/ArticlesPublications/Documents/selfdriving-cars-next-revolution. pdf,(consulted on April 7th 2014)*, 2012. (Cited on page 2.)

[138] Barry Brian Werger. Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams. *Artificial Intelligence*, 110(2):293–320, 1999. (Cited on page 7.)

[139] Wikipedia. Traffic — wikipedia, the free encyclopedia, 2014. [Online; accessed 17-July-2014]. (Cited on page 17.)

[140] Tichakorn Wongpiromsarn, Tawit Uthaicharoenpong, Yu Wang, Emilio Frazzoli, and Danwei Wang. Distributed traffic signal control for maximum network throughput. In *Proceedings of the 15th international IEEE conference on intelligent transportation systems*, pages 588–595, 2012. (Cited on pages 11, 12 and 107.)

[141] Jung-Min Yang and Jong-Hwan Kim. Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *Robotics and Automation, IEEE Transactions on*, 15(3):578–587, 1999. (Cited on pages 55 and 83.)

[142] I. Zohdy and H. Rakha. Optimizing driverless vehicles at intersections. In *10th ITS World Congress Vienna, Austria*, October 2012. (Cited on page 4.)

Appendix

Appendix A

Appendix of Chapter 2

A.1 Topology results

In the sequel, given a topological space X and a subset $C \subset X$, C^c denotes the complementary of a C in X, $\mathrm{cl}\,(C)$ its closure and ∂C its boundary.

Lemma A.1.1. *Let A and B be two disjoint subsets of a topological space X ($A \cap B = \emptyset$). Let $f : [0,1] \to A \cup B$ denote a continuous application taking values in $A \cup B$ and satisfying $f(0) \in A$ and $f(1) \in B$. Then there exists $t^0 \in [0,1]$ such that $f(t^0) \in \partial A \cap \partial B$.*

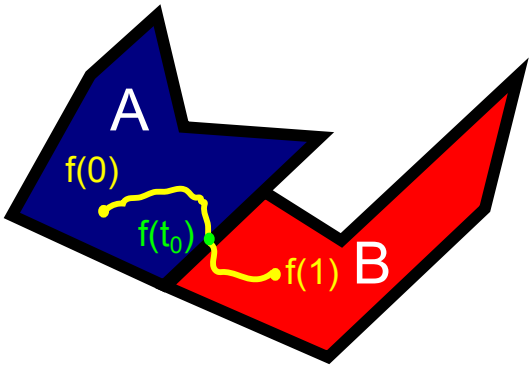


Figure A.1: Illustration of Lemma A.1.1

Proof. Consider t^0 defined below:

$$\begin{aligned} t^0 &:= \sup\{t \in [0,1] : f(t) \in A\} \\ &= \sup\{t \in [0,1] : f(t) \notin B\} \end{aligned} \tag{A.1}$$

It exists as the supremum of a non-empty ($f(0) \in A$) upper-bounded subset of $\mathbb{R}$.

Take some $r > 0$ and let $\mathcal{B}(f(t^0), r)$ denote the open ball of radius r centered in $f(t^0)$. Consider two options:

- either $t^0 = 1$, and $\mathcal{B}(f(t^0), r) \cap B \neq \emptyset$ as $f(1) \in B$;
- or $t^0 < 1$, then, by continuity of f, for small enough $\eta > 0$, we have $f(t^0 + \eta) \in \mathcal{B}(f(t^0), r)$ and $f(t^0 + \eta) \in B$ as $t^0 + \eta > t^0$ by definition of t^0.

In both cases, we have $\mathcal{B}(f(t^0), r) \cap B \neq \emptyset$. Again, consider two options:

- either $t^0 = 0$, and $\mathcal{B}(f(t^0), r) \cap A \neq \emptyset$ as $f(0) \in A$;
- or $t^0 > 0$, then, by continuity of f, for small enough $\eta > 0$, we have for all $t \in (t^0 - \eta, t^0]$, $f(t) \in \mathcal{B}(f(t^0), r)$ and there exists some $t^1 \in (t^0 - \eta, t^0]$ such that $f(t^1) \in A$ by definition of t^0 (otherwise, t^0 would not be the supremum defined in Equation (A.1)).

In both cases, we have $\mathcal{B}(f(t^0), r) \cap A \neq \emptyset$.

In conclusion, for all $r > 0$, we have:

$$\mathcal{B}(f(t^0), r) \cap A \neq \emptyset \tag{A.2}$$
$$\mathcal{B}(f(t^0), r) \cap B \neq \emptyset \tag{A.3}$$

Let C^c denote the complementary of a subset C of X. Since $A \cap B = \emptyset$, we have $B \subset A^c$ and $A \subset B^c$, so that we also have:

$$\mathcal{B}(f(t^0), r) \cap B^c \neq \emptyset \tag{A.4}$$
$$\mathcal{B}(f(t^0), r) \cap A^c \neq \emptyset \tag{A.5}$$

By definition of the boundary, as Equations (A.2) and (A.5) are satisfied for all $r > 0$, we have $f(t^0) \in \partial A$ and as Equations (A.3) and (A.4) are satisfied for all $r > 0$, we have $f(t^0) \in \partial B$. In conclusion, we have:

$$f(t^0) \in \partial A \cap \partial B \tag{A.6}$$

□

Lemma A.1.2. *Let A and B be two subsets of a topological space X and assume that B is open, then we have:*

$$\partial (A \setminus B) \cap B = \emptyset \tag{A.7}$$

Proof. By simple manipulations, we obtain:

$$\partial (A \setminus B) \subset \mathrm{cl}\,(A \setminus B) = \mathrm{cl}\,(A \cap B^c) \subset \mathrm{cl}\,(B^c) \tag{A.8}$$

As B is open, B^c is closed, so that:

$$\partial (A \setminus B) \subset B^c \tag{A.9}$$

As a consequence,

$$(\partial (A \setminus B) \cap B) \subset (B \cap B^c) = \emptyset \tag{A.10}$$

□

A.2 Proof of the South-West North-East completion

First of all, note that the following property is a direct consequence of the non-decreasing constraint.

Property A.2.1. *For all* $\varphi \in \Phi(\chi^{\text{free}})$, $t^0 \in [0,1]$, *we have:*

$$\forall t \in [0,1], \varphi(t) \in \left(\varphi(t^0) - \mathbb{R}_+^{*n}\right) \cup \left(\varphi(t^0) + \mathbb{R}_+^n\right) \tag{A.11}$$

Proof of Lemma 2.2.1. We will prove Lemma 2.2.1 by contradiction. Take $i, j \in \mathcal{R}$, $x^0 \in \left(\chi_{i\succ j}^{\text{obs}} \cap \chi_{j\succ i}^{\text{obs}}\right)$ and assume there exists a feasible path $\varphi \in \Phi(\chi^{\text{free}})$ such that for some $t^0 \in [0,1]$, $\varphi(t^0) = x^0$. Define A and B as follows:

$$A := x^0 - \mathbb{R}_+\mathbf{e}_j + \mathbb{R}_+\mathbf{e}_i \tag{A.12}$$
$$B := x^0 - \mathbb{R}_+\mathbf{e}_i + \mathbb{R}_+\mathbf{e}_j \tag{A.13}$$

As $x^0 \in \left(\chi_{i\succ j}^{\text{obs}} \cap \chi_{j\succ i}^{\text{obs}}\right)$, we have:

$$\chi_{ij}^{\text{obs}} \cap A \neq \emptyset \tag{A.14}$$
$$\chi_{ij}^{\text{obs}} \cap B \neq \emptyset \tag{A.15}$$

Take $x^1 \in \chi_{ij}^{\text{obs}} \cap A$ and $x^2 \in \chi_{ij}^{\text{obs}} \cap B$. By construction of A and B, the following inequalities hold:

$$x_i^2 \leq x_i^0 \leq x_i^1 \tag{A.16}$$
$$x_j^1 \leq x_j^0 \leq x_j^2 \tag{A.17}$$

By convexity of χ_{ij}^{obs} and as χ_{ij}^{obs} is a cylinder,

$$\Sigma := \left\{ x \in \chi : \begin{matrix} x_i = \alpha x_i^1 + (1-\alpha)x_i^2 \\ x_j = \alpha x_j^1 + (1-\alpha)x_j^2 \end{matrix}, \alpha \in [0,1] \right\} \tag{A.18}$$

is a subset of χ_{ij}^{obs}. Define K^+ and K^- as follows:

$$K^+ := \Sigma + \mathbb{R}_+\mathbf{e}_i + \mathbb{R}_+\mathbf{e}_j \tag{A.19}$$
$$K^- := \Sigma - \mathbb{R}_+^*\mathbf{e}_i - \mathbb{R}_+^*\mathbf{e}_j \tag{A.20}$$

Note that $K^+ \cap K^- = \emptyset$ and $\partial K^+ \cap \partial K^- = \Sigma$. By construction, we have:

$$\left(\varphi(t^0) + \mathbb{R}_+^n\right) \subset (K^+ \cup K^-) \tag{A.21}$$
$$\left(\varphi(t^0) - \mathbb{R}_+^{*n}\right) \subset (K^+ \cup K^-) \tag{A.22}$$

Hence, by Property A.2.1, we have for all $t \in [0,1]$,

$$\varphi(t) \in \left(K^+ \cup K^-\right) \tag{A.23}$$

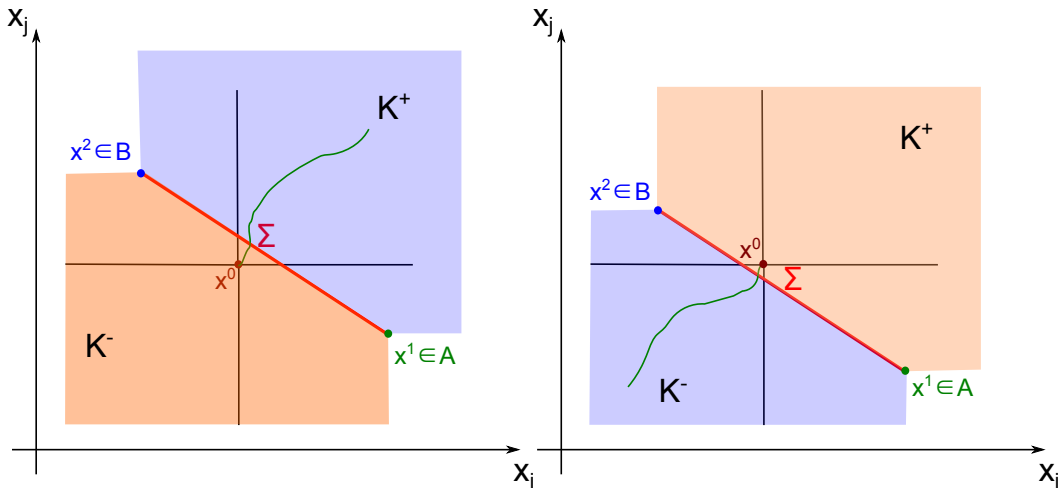


Figure A.2: The two cases that appear in the proof of Lemma 2.2.1

By construction, we have two options as depicted in Figure A.2:

- either $x^0 \in K^+$ (right drawing of Figure A.2). This case can be interpreted as the case when x^0 is not "reachable". As $(\varphi(0) - \mathbb{R}^n_+) \subset \chi^{\text{free}}$ and $\varphi(0) \in (\varphi(t^0) - \mathbb{R}^n_+)$, we have:

$$\varphi(0) \in K^- \tag{A.24}$$
$$\varphi(t^0) \in K^+ \tag{A.25}$$

  Hence, as φ is continuous, by Lemma A.1.1 (see Appendix A.1), there exists some t^Σ such that $\varphi(t^\Sigma) \in (\partial K^- \cap \partial K^+) = \Sigma \subset \chi^{\text{obs}}_{ij}$ and φ is not collision-free.

- or $x^0 \in K^-$ (left drawing of Figure A.2). This case can be interpreted as the case when x^0 will inevitably lead to a deadlock between robots i and j. As $(\varphi(1) + \mathbb{R}^n_+) \subset \chi^{\text{free}}$ and $\varphi(1) \in (\varphi(t^0) + \mathbb{R}^n_+)$, we have:

$$\varphi(1) \in K^+ \tag{A.26}$$
$$\varphi(t^0) \in K^- \tag{A.27}$$

  Hence, as φ is continuous, by Lemma A.1.1 (see Appendix A.1), there exists some t^Σ such that $\varphi(t^\Sigma) \in (\partial K^+ \cap \partial K^-) = \Sigma \subset \chi^{\text{obs}}_{ij}$ and φ is not collision-free.

In conclusion, there is no feasible path going through any configuration $x^0 \in \chi^{\text{obs}}_{i \succ j} \cap \chi^{\text{obs}}_{j \succ i}$. □

A.3 Proof of the existence of paths with maximal margin

Lemma A.3.1 (Existence of paths with maximal margin)**.** *Given a priority graph $G \in \mathcal{G}$, consider ρ_G defined as follows:*

$$\rho_G := \begin{cases} \sup\left\{r \geq 0 : \Phi(\chi_G^{\text{free}} \ominus [-r,r]^n) \neq \emptyset\right\} & \text{if } \Phi(\chi_G^{\text{free}}) \neq \emptyset \\ -\inf\left\{r > 0 : \Phi(\chi_G^{\text{free}} + [-r,r]^n) \neq \emptyset\right\} & \text{else.} \end{cases} \tag{A.28}$$

$\rho_G \in \mathbb{R} \cup \{+\infty\}$ and if $\rho_G \in \mathbb{R}$, it is attained, so that we can use the maximal element notation:

$$\rho_G := \begin{cases} \max\left\{r \geq 0 : \Phi(\chi_G^{\text{free}} \ominus [-r,r]^n) \neq \emptyset\right\} & \text{if } \Phi(\chi_G^{\text{free}}) \neq \emptyset \\ -\min\left\{r > 0 : \Phi(\chi_G^{\text{free}} + [-r,r]^n) \neq \emptyset\right\} & \text{else.} \end{cases} \tag{A.29}$$

Proof of $\rho_G \in \mathbb{R} \cup \{+\infty\}$. Consider a path $\varphi \in \Phi(\chi)$ whose image is the segment joining $\underline{x}^{\text{obs}}$ to $\overline{x}^{\text{obs}}$. For big enough enough $r \in \mathbb{R}_+$, this segment is collision-free with regards to $\chi_G^{\text{obs}} \ominus [-r,r]^n$. Moreover, $(\underline{x}^{\text{obs}} - \mathbb{R}_+^n) \subset \chi_G^{\text{free}} \subset \chi_G^{\text{free}} + [-r,r]^n$ and $(\overline{x}^{\text{obs}} + \mathbb{R}_+^n) \subset \chi_G^{\text{free}} \subset \chi_G^{\text{free}} + [-r,r]^n$, so that $\varphi \in \Phi(\chi_G^{\text{free}} + [-r,r]^n)$. Hence, $\rho_G \neq -\infty$ and we obtain $\rho_G \in \mathbb{R} \cup \{+\infty\}$. □

Proof that it is attained. Take a priority graph and assume that $\rho_G \in \mathbb{R}$. Define $C_G^{\text{free}} := \chi_G^{\text{free}} \ominus (-\rho_G, \rho_G)^n$ if $\rho_G \geq 0$ (resp. $C_G^{\text{free}} := \chi_G^{\text{free}} + (-|\rho_G|, |\rho_G|)^n$ if $\rho_G \leq 0$) and $C_G^{\text{obs}} := \chi \setminus C_G^{\text{free}}$. We have to prove that:

$$\Phi(C_G^{\text{free}}) \neq \emptyset \tag{A.30}$$

We have the following identity:

$$C_G^{\text{free}} = \bigcap_{r>0} \left(C_G^{\text{free}} + [-r,r]^n\right) \tag{A.31}$$

Hence, $\Phi(C_G^{\text{free}})$ can be expressed as the limit of a nested sequence of sets of paths as follows:

$$\Phi(C_G^{\text{free}}) = \bigcap_{r>0} \Phi\left(C_G^{\text{free}} + [-r,r]^n\right) \tag{A.32}$$

By definition of ρ_G, we have $\Phi\left(C_G^{\text{free}} + [-r,r]^n\right) \neq \emptyset$ for all $r > 0$, so that $(\Phi\left(C_G^{\text{free}} + [-r,r]^n\right))_{r>0}$ is a nested sequence of non-empty sets of paths whose limit is $\Phi(C_G^{\text{free}})$. We have to prove that this limit is not the empty set.

Consider the topology of pointwise convergence on $\Phi(\chi)$. Define $\overline{\underline{\Phi}}(\chi) \subset \Phi(\chi)$ as follows:

$$\overline{\underline{\Phi}}(\chi) := \left\{\varphi \in \Phi(\chi) : \forall t \in [0,1], \underline{x}^{\text{obs}} \leq \varphi(t) \leq \overline{x}^{\text{obs}}\right\} \tag{A.33}$$

As for all $\varphi \in \overline{\underline{\Phi}}(\chi)$ and $i \in \mathcal{R}$, $\text{Im}\,(\varphi_i) \subset [\underline{x}_i^{\text{obs}}, \overline{x}_i^{\text{obs}}]$, by Tychonoff's theorem, $\overline{\underline{\Phi}}(\chi)$ with the topology of pointwise convergence is compact in $\Phi(\chi)$. Intersecting with $\overline{\underline{\Phi}}(\chi)$ in (A.32), we obtain:

$$\Phi(C_G^{\text{free}}) \cap \overline{\underline{\Phi}}(\chi) = \bigcap_{r>0} \left(\Phi\left(C_G^{\text{free}} + [-r,r]^n\right) \cap \overline{\underline{\Phi}}(\chi)\right) \tag{A.34}$$

Now, we are going to use Cantor's intersection theorem to prove that $\Phi(C_G^{\text{free}}) \cap \overline{\underline{\Phi}}(\chi) \neq \emptyset$. To this purpose, we are going to prove that for all $r > 0$, $\Phi\left(C_G^{\text{free}} + [-r,r]^n\right) \cap \overline{\underline{\Phi}}(\chi)$ is a non-empty compact in $\Phi(\chi)$. As $\overline{\underline{\Phi}}(\chi)$ is compact, it is sufficient to prove that $\Phi\left(C_G^{\text{free}} + [-r,r]^n\right) \cap \overline{\underline{\Phi}}(\chi)$ is (a) non-empty and (b) closed in $\Phi(\chi)$.

(a) $\Phi\left(C_G^{\text{free}} + [-r,r]^n\right)$ is non-empty. Moreover, taking a path $\varphi \in \Phi\left(C_G^{\text{free}} + [-r,r]^n\right)$, and building the path $\tilde{\varphi} := \min(\max(\underline{x}^{\text{obs}}, \varphi), \overline{x}^{\text{obs}})$ yields a path in $\overline{\underline{\Phi}}(\chi) \cap \Phi\left(C_G^{\text{free}} + [-r,r]^n\right)$. As a result, $\Phi\left(C_G^{\text{free}} + [-r,r]^n\right) \cap \overline{\underline{\Phi}}(\chi)$ is not empty.

(b) First, we prove that the complementary of $\Phi\left(C_G^{\text{free}} + [-r,r]^n\right)$ in $\Phi(\chi)$ is open. Take a path $\varphi \in \Phi(\chi)$ and assume $\varphi \notin \Phi\left(C_G^{\text{free}} + [-r,r]^n\right)$. As $\varphi \in \Phi(\chi)$, φ is necessarily non-decreasing, so that $\varphi \notin \Phi\left(C_G^{\text{free}} + [-r,r]^n\right)$ means that φ (or $\varphi(0) - \mathbb{R}_+^n$, or $\varphi(1) + \mathbb{R}_+^n$) is not collision-free with respect to $C_G^{\text{obs}} \ominus [-r,r]^n$, which is an open set. Hence, any path $\psi \in \Phi(\chi)$ close enough to φ (in the topology of pointwise convergence) also interects $C_G^{\text{obs}} \ominus [-r,r]^n$, so that $\psi \notin \Phi\left(C_G^{\text{free}} + [-r,r]^n\right)$. It results that $\Phi\left(C_G^{\text{free}} + [-r,r]^n\right)$ is closed in $\Phi(\chi)$ (as its complementary is open). Moreover, $\overline{\underline{\Phi}}(\chi)$ is also closed as it is compact. In conclusion, $\Phi\left(C_G^{\text{free}} + [-r,r]^n\right) \cap \overline{\underline{\Phi}}(\chi)$ is closed in $\Phi(\chi)$ as the intersection of closed sets in $\Phi(\chi)$.

Applying Cantor's intersection theorem, we obtain that $\Phi(C_G^{\text{free}}) \cap \overline{\underline{\Phi}}(\chi)$ is a non-empty compact of $\Phi(\chi)$, which implies that $\Phi(C_G^{\text{free}}) \neq \emptyset$. □

A.4 Proof of the characterization of feasible priority graphs

Theorem 3.2.2 is a direct consequence of the following lemmas. We let $C_G^{\text{obs}} := \chi_G^{\text{obs}} + [-r,r]^n$ (or $C_G^{\text{obs}} := \chi_G^{\text{obs}} \ominus [-r,r]^n$) with $C_{i\succ j}^{\text{obs}} := \chi_{i\succ j}^{\text{obs}} + [-r,r]^n$ (resp. $C_{i\succ j}^{\text{obs}} := \chi_{i\succ j}^{\text{obs}} \ominus [-r,r]^n$) and we let $C_G^{\text{free}} := \chi \setminus \chi_G^{\text{obs}}$. Note that C_G^{obs} satisfy the same invariance properties as χ_G^{obs} (Properties 2.2.1, 2.2.2, 2.2.3 and 2.2.3). Lemma A.4.1 provides a necessary condition for the existence of a feasible path taking values in C_G^{free} and Lemma A.4.3 provides a sufficient condition for the existence of such path. Note that the lemmas apply in particular for $C_G^{\text{obs}} \equiv \chi_G^{\text{obs}}$ ($r = 0$).

Lemma A.4.1. *Given a priority graph $G \in \mathcal{G}$, a necessary condition for $\Phi(C_G^{\text{free}}) \neq \emptyset$ is:*

$$\forall \mathcal{C} \in \text{cycles}(G), \quad \bigcap_{(i,j)\in E(\mathcal{C})} C_{i\succ j}^{\text{obs}} = \emptyset \tag{A.35}$$

The proof of the above lemma is omitted as it is exactly the same as the proof of the necessary condition of Theorem 3.2.2 where C_G^{obs} replaces χ_G^{obs}.

In order to provide a constructive proof of the existence of feasible paths taking values in C_G^{free} under certain conditions, we introduce the concept of local priority graph (an extension of the concept introduced in the corpus of the manuscript with C_G^{obs} instead of χ_G^{obs}). Given a radius $r > 0$ and a configuration $x \in \chi$, the local priority graph with regards to C_G^{obs} at configuration x with radius $r \geq 0$ is the sub-graph $G_{|C_G^{\text{obs}},x,r}$ of G with the same vertices and whose edge set is defined below:

$$E(G_{|C_G^{\text{obs}},x,r}) := \left\{(i,j) \in E(G) : x \in \left(C_{i\succ j}^{\text{obs}} + [-r,r]^n\right)\right\} \tag{A.36}$$

Note that $G_{|x,r} \equiv G_{|\chi_G^{\text{obs}},x,r}$ which is simply referred as the priority graph at configuration x with radius $r \geq 0$ (without mentioning χ_G^{obs}).

Lemma A.4.2 (Sufficient condition for locally acyclic priority graph)**.** *Consider a priority graph $G \in \mathcal{G}$ satisfying for all elementary cycles $\mathcal{C}$ in G:*

$$\bigcap_{(i,j)\in E(\mathcal{C})} \left(C_{i\succ j}^{\text{obs}} + (-\varepsilon,\varepsilon)^n\right) = \emptyset \tag{A.37}$$

for some $\varepsilon > 0$, then $G_{|C_G^{\text{obs}},x,\varepsilon}$ is acyclic at all configurations $x \in \chi$.

The proof of the above lemma is omitted as it is exactly the same proof as Lemma 3.2.1 where C_G^{obs} replaces χ_G^{obs}.

It is of high interest to know that the local priority graph with radius $\varepsilon > 0$ with regards to C_G^{obs} is acyclic at all configurations $x \in \chi$. Indeed, when this condition is satisfied, whatever the current configuration $x \in C_G^{\text{free}}$ of the system, it is always possible to find a robot $i \in \mathcal{R}$ which can move forward the distance $\varepsilon > 0$ without colliding, which enables to construct a feasible path in C_G^{free} by iterations.

Lemma A.4.3. *Given a priority graph $G \in \mathcal{G}$, a sufficient condition for $\Phi(C_G^{\text{free}}) \neq \emptyset$ is:*

$$\exists \varepsilon > 0 : \forall \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} (C_{i\succ j}^{\text{obs}} + (-\varepsilon,\varepsilon)^n) = \emptyset \tag{A.38}$$

The proof of the above lemma is omitted as it is exactly the same proof as Lemma 3.2.1 where C_G^{obs} replaces χ_G^{obs}.

Proof of Theorem 3.2.2. We will prove Theorem 3.2.2 by combining the above lemmas. Take a priority graph $G \in \mathcal{G}$ and consider two options.

- If G satisfies:

$$\exists r > 0 : \forall \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} (\chi_{i\succ j}^{\text{obs}} + [-r,r]^n) = \emptyset \tag{A.39}$$

Then, by Lemma A.4.3, $\Phi(\chi_G^{\text{free}}) \neq \emptyset$, so G is feasible. Moreover, take $\overline{r}$ defined as follows:

$$\overline{r} := \sup\Big\{r > 0 : \forall \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} (\chi_{i\succ j}^{\text{obs}} + [-r,r]^n) = \emptyset\Big\} \quad \text{(A.40)}$$

$\overline{r}$ exists in $\mathbb{R}_+ \cup \{+\infty\}$ as the supremum of a non-empty subset of $\mathbb{R}$ (it is non-empty as (A.39) is satisfied). By Lemma A.4.1, for all $r > \overline{r}$, $\Phi(\chi_G^{\text{free}} \ominus [-r,r]^n) = \emptyset$. And by Lemma A.4.3, for all $r < \overline{r}$, $\Phi(\chi_G^{\text{free}} \ominus [-r,r]^n) \neq \emptyset$ (take $0 < \varepsilon < \overline{r} - r$). By definition of ρ_G, we have $\rho_G = \overline{r}$, i.e.,

$$\rho_G = \max\left\{r > 0 : \forall \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} (\chi_{i\succ j}^{\text{obs}} + [-r,r]^n) = \emptyset\right\} \quad \text{(A.41)}$$

where max replaces sup since $\Phi(\chi_G^{\text{free}} \ominus (-\rho_G, \rho_G)^n) \neq \emptyset$ (see Lemma A.3.1) implies $\bigcap_{(i,j)\in E(\mathcal{C})}(\chi_{i\succ j}^{\text{obs}} + (-\rho_G, \rho_G))^n) = \emptyset$ by Lemma A.4.3.

- If G satisfies:

$$\forall r > 0 : \exists \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} (\chi_{i\succ j}^{\text{obs}} + [-r,r]^n) \neq \emptyset \quad \text{(A.42)}$$

Then, by Lemma A.4.1, $\rho_G \leq 0$ (G is not feasible or it is feasible with a safety margin of 0). Take $\overline{r}$ defined as follows:

$$\overline{r} := \inf\Big\{r \geq 0 : \forall \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} (\chi_{i\succ j}^{\text{obs}} \ominus [-r,r]^n) = \emptyset\Big\} \quad \text{(A.43)}$$

By Lemma A.4.3, for all $r > \overline{r}$, $\Phi(\chi_G^{\text{free}} + [-r,r]^n) \neq \emptyset$ (take $0 < \varepsilon < \overline{r} - r$) and by Lemma A.4.1, for all $r < \overline{r}$, $\Phi(\chi_G^{\text{free}} + [-r,r]^n) = \emptyset$. By definition of ρ_G, we have $\rho_G = -\overline{r}$, i.e.,

$$\rho_G = -\min\left\{r \geq 0 : \exists \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} (\chi_{i\succ j}^{\text{obs}} \ominus [-r,r]^n) = \emptyset\right\} \quad \text{(A.44)}$$

where min replaces inf and $\geq$ replaces $>$ as $\Phi(\chi_G^{\text{free}} \ominus (-\rho_G, \rho_G)^n) \neq \emptyset$ (see Lemma A.3.1) implies $\bigcap_{(i,j)\in E(\mathcal{C})}(\chi_{i\succ j}^{\text{obs}} \ominus (-\rho_G, \rho_G)^n) = \emptyset$ by Lemma A.4.3. Hence, in that case, G is feasible if and only if $\bigcap_{(i,j)\in E(\mathcal{C})} \chi_{i\succ j}^{\text{obs}} = \emptyset$.

In conclusion, a necessary and sufficient condition for G being feasible is:

$$\forall \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} \chi_{i\succ j}^{\text{obs}} = \emptyset \quad \text{(A.45)}$$

and if this condition is satisfied, the feasibility margin is given by:

$$\rho_G = \max\left\{r \geq 0 : \forall \mathcal{C} \in \text{cycles}(G), \bigcap_{(i,j)\in E(\mathcal{C})} (\chi_{i\succ j}^{\text{obs}} + [-r,r]^n) = \emptyset\right\} \quad \text{(A.46)}$$

where $\geq$ replaces $>$ of Equation (A.41) to include the case $\rho_G = 0$. □

Appendix B

Priority preserving control: extension to feasible cyclic priorities

The results presented in Part II have been proved under the assumption that the assigned priorities are acyclic. However, they can be extended to all feasible priority graphs – including those containing cycles – under certain mild conditions. This is the topic of this appendix chapter.

Sketch of the chapter Sections B.1, B.2 and B.3 respectively extend the results of Chapter 4 (control in velocity), 5 (control in acceleration) and 6 (bounded uncertainty).

B.1 In the absence of inertia

First of all, we examine the case of robots controlled in velocity along their paths. In Chapter 4, the acyclicity of G is key as it is even necessary to the existence of the control law (see Theorem 4.2.1). A relaxed sufficient condition on the priority graph for all the results of Chapter 4 to hold is that the maximum distance traveled by a robot in one time slot satisfies:

$$\max_{i \in \mathcal{R}} \overline{v}_i \leq \rho_G \tag{B.1}$$

In particular, it is necessary for the safety margin to be strictly positive. In practice, to satisfy this assumption, one can choose a sufficiently small time slot length (the maximum distance traveled by a robot in one time slot is proportional to the time slot length). However, due to actuators limitations, there is a lower bound in time slot length. Assume that the minimal time slot length is chosen. A robot i either is stopped or travels the distance $\overline{v}_i$ which can be seen as the control resolution. Basically, Equation (B.1) states that the control resolution needs to be lower than the safety margin ρ_G.

All the results of Section B.1 (existence, priority preservation, optimality, liveness) are extended under Condition (B.1) and proofs are provided. The key idea is to use the fact that if the priority graph is feasible with a strictly positive margin ρ_G, then the local priority graph is acyclic at all configurations. Hence, it is always possible to move forward some robot provided the control resolution is small enough. The evolution of the multi robot system under control law f^G with cyclic priorities is depicted in Figure B.1 (the evolution of the local priority graph is also represented). Figure B.2 depicts the trajectory in the coordination space.

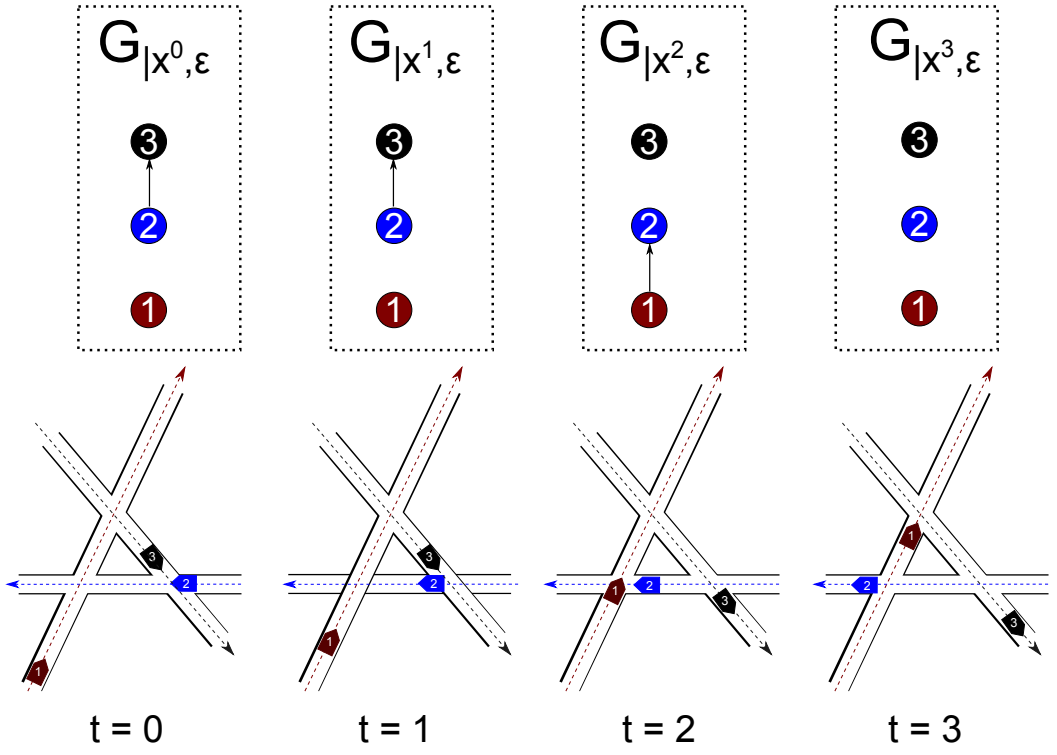


Figure B.1: A three-robot system with cyclic assigned priorities: $1 \succ 2$, $2 \succ 3$, *and* $3 \succ 1$. *Robots are controlled under control law* f^G. *The top drawings show the evolution of the local priority graph and the bottom drawings depict the evolution of the robots along their paths.*

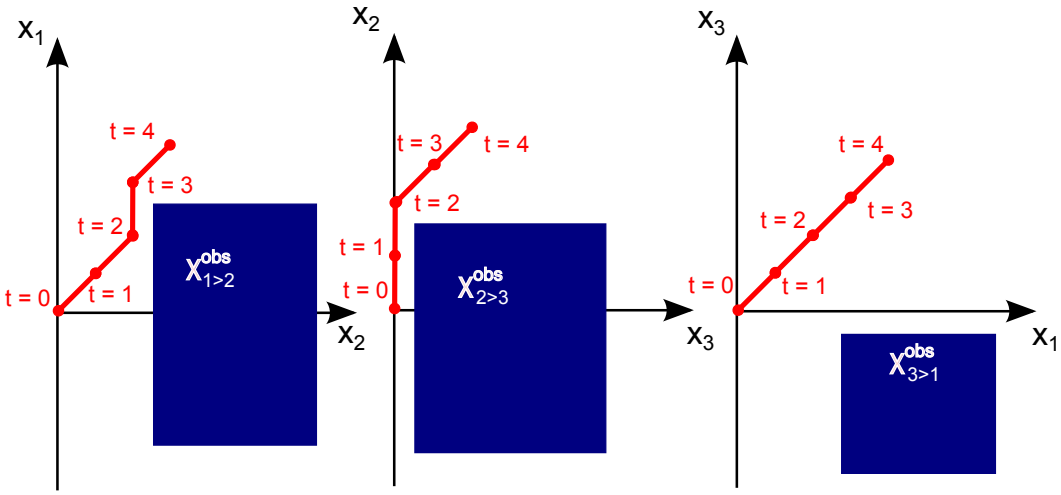


Figure B.2: Trajectory in the coordination space for the three-robot system under control law f^G *of Figure 4.2.*

B.1.1 Existence

Theorem B.1.1 (Control law existence)**.** *Given a feasible priority graph G with a strictly positive feasibility margin satisfying Condition (B.1), Equation (4.5) uniquely defines a control law $f^G : \chi \to V$.*

Proof. Take a feasible priority graph satisfying Condition (B.1). By Lemma 3.2.1, $G_{|x,\rho_G}$ is acyclic at all configurations $x \in \chi$. Moreover, by definition of the local priority graph and as Condition (B.1) is satisfied, Equation (4.5) is equivalent to:

$$f_i^G(x) := \begin{cases} 0 & \text{if } \exists (j,i) \in E(G_{|x,\rho_G}), t \in [0,1] \text{ s.t. } (x + t\,(\overline{v}_i \mathbf{e}_i + f_j(x)\mathbf{e}_j)) \in \chi^{\text{obs}}_{j \succ i} \\ \overline{v}_i & \text{else.} \end{cases} \tag{B.2}$$

Since $G_{|x,\rho_G}$ is a directed acyclic graph, there exists a topological ordering of the graph such that for every edge $(j,i) \in E(G_{|x,\rho_G})$, j comes before i in the ordering. Hence, following the topological order induced by $G_{|x,\rho_G}$, it is possible to compute $f_i^G(x)$ for all $i \in \mathcal{R}$ iteratively. As a result, Equation (B.2) uniquely defines a control law $f^G : \chi \to V$. □

B.1.2 Safety

Theorem B.1.2 (Priority preservation)**.** *Given a feasible priority graph G with a strictly positive feasibility margin satisfying Condition (B.1), χ_G^{free} is positively invariant for the system under control law f^G:*

$$\forall x \in \chi_G^{\text{free}}, \forall t \geq 0, \phi(t, x, f^G) \in \chi_G^{\text{free}} \tag{B.3}$$

The proof is omitted as it is exactly the same as the proof of Theorem 4.2.2. Indeed, in the proof of Theorem 4.2.2, the acyclicity of the priority graph is useful only to ensure that the control law f_G is well defined. Condition (B.1) precisely ensures that the control law f_G is well defined even if priorities are cyclic (see Theorem B.1.1).

B.1.3 Optimality

Theorem B.1.3 (Optimality)**.** *Given a feasible priority graph G with a strictly positive feasibility margin satisfying Condition (B.1), the control law f^G is optimal for the priority graph G.*

The proof is omitted as it is exactly the same as the proof of Theorem 4.2.3. Again, in the proof of Theorem 4.2.3 is useful only to ensure that the control law f_G is well defined which is already guaranteed by Condition (B.1).

B.1.4 Liveness

Theorem B.1.4 (Liveness)**.** *Given a feasible priority graph G with a strictly positive feasibility margin satisfying Condition (B.1) and a configuration $x^0 \in \chi_G^{\text{free}}$, there exists $T > 0$ such that:*

$$\phi(T, x^0, f^G) \in \chi^{\text{goal}} \tag{B.4}$$

Proof. Take a feasible priority graph G satisfying Condition (B.1). First of all, note that by Lemma 3.2.1, the local priority graph of radius $\rho_G > 0$ is acyclic at all configurations. Take a configuration $x^0 \in \chi_G^{\text{free}}$. Define the set $I(t) \subset \mathcal{R}$ defined as follows for all $t \geq 0$:

$$I(t) := \left\{ i \in \mathcal{R} : \phi_i(t, x^0, f^G) < \overline{x}_i^{\text{obs}} \right\} \tag{B.5}$$

Theorem B.1.4 is equivalent to the existence of $T > 0$ such that:

$$I(T) = \emptyset \tag{B.6}$$

First of all, note that the set $I(t)$ decreases through time, as the trajectory of robots is non-decreasing. At every time slot k, if $I(k)$ is not empty, consider i^*, a maximal element of the (acyclic) sub-graph of $G_{|\phi(k,x^0,f^G),\rho_G}$ retaining only vertices in $I(k)$. Remember that, by definition of the local priority graph and as Condition (B.1) is satisfied, the definition of the control law is equivalent to:

$$f_i^G(x) = \begin{cases} 0 & \text{if } \exists (j,i) \in E(G_{|x,\rho_G}), t \in [0,1] \text{ s.t. } (x + t\,(\overline{v}_i \mathbf{e}_i + f_j(x)\mathbf{e}_j)) \in \chi_{j \succ i}^{\text{obs}} \\ \overline{v}_i & \text{else.} \end{cases} \tag{B.7}$$

Hence, by construction of i^*, we have $f_{i^*}^G(\phi(k, x^0, f^G)) = \overline{v}_{i^*}$, so that:

$$\phi_{i^*}(k+1, x^0, f^G) - \phi_{i^*}(k, x^0, f^G) = \overline{v}_{i^*} \tag{B.8}$$

Hence, at every time slot k, there is at least one robot $i^* \in I(k)$ traveling distance $\overline{v}_{i^*}$. As there is initially a finite number of robots in $I(0)$ and as each robot i has a finite distance to travel to reach position $\overline{x}_i^{\text{obs}}$, it is guaranteed that $I(t)$ gets empty in finite time, i.e., the multi robot system reaches χ^{goal} in finite time. □

B.2 Under kinodynamic constraints

When robots are controlled in acceleration, the control law g^G proposed in Chapter 5 is well defined, safe and robust even if the priority graph G is not feasible. However, the liveness guarantee of Theorem 5.2.3 has been proved under the assumption that priorities are acyclic. Actually, liveness is still guaranteed under a less restrictive

assumption on the priority graph. Again, the assumption is on the safety margin of the priority graph:

$$\max_{i\in\mathcal{R}} \sup_{t\in\mathbb{R}_+} \phi_{x,i}(t, 0, \mathbf{u}_i^{\text{impulse}}) \leq \rho_G \tag{B.9}$$

Basically, under control in acceleration, $\sup_{t\in\mathbb{R}_+} \phi_{x,i}(t, 0, \mathbf{u}_i^{\text{impulse}})$ represents the control resolution as it is the maximum distance traveled by a robot starting from stop and applying an impulse control considering all robots. Again, Equation (B.9) states that the control resolution needs to be lower than the safety margin. Liveness under Condition (B.9) is proved in the sequel.

Theorem B.2.1 (Liveness)**.** *Given a feasible priority graph G with a strictly positive feasibility margin satisfying Condition (B.9) and an initial brake safe state $s \in B_G$, there exists $T > 0$ such that:*

$$\phi_x(T, s, g^G) \in \chi^{\text{goal}} \tag{B.10}$$

We start with some lemmas.

Lemma B.2.1. *Given $i \in \mathcal{R}$, $s_i \in S_i$ and $\mathbf{u}_i \in \mathbf{U}_i$ taking values in $\{\underline{u}_i, \overline{u}_i\}$, the following implication holds:*

$$\left[\forall k \in \mathbb{N}, \phi_{x,i}(k, s_i, \mathbf{u}_i) \leq \overline{x}_i^{\text{obs}}\right] \Rightarrow \left[\exists k^0 \in \mathbb{N} : \forall k \geq k^0, \mathbf{u}_i(k) = \underline{u}_i\right] \tag{B.11}$$

Proof. We reason by contraposition. Consider $i \in \mathcal{R}$, $s_i \in S_i$ and $\mathbf{u}_i \in \mathbf{U}_i$ taking values in $\{\underline{u}_i, \overline{u}_i\}$. Assume that $\forall k^0 \in \mathbb{N}, \exists k \in \mathbb{N}, k \geq k^0 : \mathbf{u}_i(k) = \overline{u}_i$. Then, we can build an infinite sequence of slots $\{k_1^0, k_2^0, ...\}$ where $\mathbf{u}_i$ takes value $\overline{u}_i$. At each of them, the increase in $\phi_{x,i}(k, s_i, \mathbf{u}_i)$ within the slot is at least of $\phi_{x,i}(1, 0, \overline{\mathbf{u}}_i) > 0$. Hence, we have $\lim_{k\to+\infty} \phi_{x,i}(k, s_i, \mathbf{u}_i) = +\infty$ and in particular $\phi_{x,i}(k, s_i, \mathbf{u}_i) > \overline{x}_i^{\text{obs}}$ for some $k \in \mathbb{N}$. □

Lemma B.2.2. *Given $i \in \mathcal{R}$, for all $s_i \in S_i$, there exists $k^1 \in \mathbb{N}$ such that:*

$$\forall k \geq k^1, \phi_{v,i}(k, s_i, \underline{\mathbf{u}}_i) = 0 \tag{B.12}$$

Proof. Take $i \in \mathcal{R}$, $s_i \in S_i$ and consider the trajectory of robot i under control $\underline{\mathbf{u}}_i$. At each time slot, there are two options:

- the velocity of robot i vanishes within the slot;

- the variation of the velocity of robot i within the slot is $\underline{u}_i < 0$.

In conclusion, the velocity of robot i vanishes in finite time. □

Lemma B.2.3. *Given a priority graph $G \in \mathcal{G}$ satisfying Condition (B.9) and a state $s \in S$, for all $i \in \mathcal{R}$ satisfying $\pi_{v,i}(s) = 0$, we have:*

$$g_i^G(s) = \begin{cases} \underline{u}_i & \text{if } \exists (j,i) \in E(G_{|\pi_x(s),\rho_G}), \exists t \geq 0 \text{ s.t. } \phi_x(t,s,\tilde{\mathbf{u}}^i) \in \chi^{\text{obs}}_{j \succ i} \\ \overline{u}_i & \text{else.} \end{cases} \tag{B.13}$$

Proof. Take a priority graph $G \in \mathcal{G}$ satisfying Condition (B.9), a state $s = (x, v) \in S$ and $i \in \mathcal{R}$ satisfying $\pi_{v,i}(s) = 0$. By construction of g^G, we have:

$$g_i^G(s) := \begin{cases} \underline{u}_i & \text{if } \exists (j,i) \in E(G), \exists t \geq 0 \text{ s.t. } \phi_x(t,s,\tilde{\mathbf{u}}^i) \in \chi^{\text{obs}}_{j \succ i} \\ \overline{u}_i & \text{else.} \end{cases} \tag{B.14}$$

Hence, we have to prove that for all $(j,i) \in E(G)$, if $\phi_x(t,s,\tilde{\mathbf{u}}^i) \in \chi^{\text{obs}}_{j \succ i}$ for some $t \geq 0$, then we have $(j,i) \in E(G_{|x,\rho_G})$. We will prove this by contraposition.

Take $(j,i) \in E(G)$ and assume that $(j,i) \notin E(G_{|x,\rho_G})$, then we have:

$$x + \rho_G \mathbf{e}_i \in \chi^{\text{free}}_{j \succ i} \tag{B.15}$$

By assumption, we have $\pi_{v,i}(s) = 0$. Hence, we obtain for all $t \geq 0$:

$$\phi_i(t,s,\tilde{u}^i) = s_i + \phi_i(t,0,\tilde{u}^i) \tag{B.16}$$

Using Condition (B.9) and as $t \mapsto \phi_{x,j}(t,s,\tilde{u}^i)$ is non-decreasing, we obtain:

$$\phi_{x,i}(t,s,\tilde{u}^i) \leq x_i + \rho_G \tag{B.17}$$

$$\phi_{x,j}(t,s,\tilde{u}^i) \geq x_j \tag{B.18}$$

By Property 2.2.1, Equation (B.15) implies that $\phi_x(t,s,\tilde{u}^i) \in \chi^{\text{free}}_{j \succ i}$ for all $t \geq 0$. □

Lemma B.2.4. *Given $s \in S$, for all $i, j \in \mathcal{R}$, for all $\mathbf{u} \in \mathbf{U}$, the following implication holds:*

$$\left[\exists t \geq 0 : \phi_x(t,s,\mathbf{u}) \in \chi^{\text{obs}}_{j \succ i}\right] \Rightarrow \left[\pi_{x,j}(s) < \overline{x}^{\text{obs}}_j\right] \tag{B.19}$$

Proof. Take $s \in S$, $i, j \in \mathcal{R}$ and $\mathbf{u} \in \mathbf{U}$. The flow $t \mapsto \phi_x(t,s,\mathbf{u})$ is non-decreasing. Hence, we have:

$$\phi_{x,j}(t,s,\mathbf{u}) \geq \phi_{x,j}(0,s,\mathbf{u}) = \pi_{x,j}(s) \tag{B.20}$$

Assume that $\phi_x(t,s,\mathbf{u}) \in \chi^{\text{obs}}_{j \succ i}$ for some $t \geq 0$. It implies that $\phi_{x,j}(t,s,\mathbf{u}) < \sup\{x_j : x \in \chi^{\text{obs}}_{j \succ i}\} \leq \overline{x}^{\text{obs}}_j$. By transitivity, we obtain $\pi_{x,j}(s) < \overline{x}^{\text{obs}}_j$. □

Proof of Theorem B.2.1. Take a priority graph $G \in \mathcal{G}$ satisfying Condition (B.9). Remember that the local priority graph with radius ρ_G is acyclic at all configuration by Lemma 3.2.1. Consider the set $I(t)$ defined as follows:

$$\forall t \in \mathbb{R}_+, I(t) := \left\{ i \in \mathcal{R} : \phi_{x,i}(t, s, g^G) < \overline{x}_i^{\text{obs}} \right\} \tag{B.21}$$

We prove Theorem B.2.1 by contradiction. Theorem B.2.1 is equivalent to the existence of $T > 0$ such that:

$$I(T) = \emptyset \tag{B.22}$$

Assume that for all $t \geq 0$, $I(t) \neq \emptyset$. Then, consider $\underline{I}$ defined as follows:

$$\underline{I} := \cap_{t \geq 0} I(t) \neq \emptyset \tag{B.23}$$

We have for all $i \in \underline{I}$,

$$\forall k \in \mathbb{N}, \phi_{x,i}(k, s, g^G) < \overline{x}_i^{\text{obs}} \tag{B.24}$$

By Lemma B.2.1, a necessary condition for the above equation to be satisfied is that for all $i \in \underline{I}$, there exists $k_i^0 \in \mathbb{N}$ such that:

$$\forall k \geq k_i^0, g_i^G(\phi(k, s, g^G)) = \underline{u}_i \tag{B.25}$$

Hence, for $k^0 := \max_{i \in \underline{I}} k_i^0$, we have:

$$\forall i \in \underline{I}, \forall k \geq k^0, g_i^G(\phi(k, s, g^G)) = \underline{u}_i \tag{B.26}$$

By Lemma B.2.2, the velocity of all robots $i \in \underline{I}$ will vanish in finite time. There exists $k^1 \geq k^0$ such that:

$$\forall k \geq k^1, g_i^G(\phi(k, s, g^G)) = \underline{u}_i \tag{B.27}$$

$$\phi_{v,i}(k, s, g^G) = 0 \tag{B.28}$$

Take $k \geq k^1$ big enough, so that $I(k) = \underline{I}$. By Lemma B.2.3, for all $i \in \underline{I}$, there exists $(j, i) \in E(G_{|\phi_x(k,s,g^G),\rho_G})$ such that $\phi_x(\mathbb{R}_+, \phi(k, s, g^G), \tilde{\mathbf{u}}^i) \cap \chi_{j \succ i}^{\text{obs}} \neq \emptyset$ and we have necessarily $\phi_{x,j}(k, s, g^G) < \overline{x}_j^{\text{obs}}$ by Lemma B.2.4, i.e., $j \in I(k) = \underline{I}$. As a result, $G_{|\phi_x(k,s,g^G),\rho_G}$ should be cyclic (there would be a cycle involving vertices in $\underline{I}$). It is absurd. □

B.3 Under uncertainty in control and sensing

Again, the control law $\hat{g}^G$ under the scenario of bounded uncertainty in control and sensing, as proposed in Chapter 6 is well defined, safe and robust even if the priority graph G is not feasible. However, if the priority graph is feasible but contains cycles, we need to make an additional assumption in order to prove liveness. First of all, we assume that the observation signal $\mathbf{y}$ is such that for all $i \in \mathcal{R}$ and $k \in \mathbb{N}$, $\mathbf{y}_i^x(k)$ is an interval of length σ_i^x. We say σ^x is the observation precision on the position. The condition to ensure liveness is:

$$\max_{i \in \mathcal{R}} \left(\sigma_i^x + \sup_{t \in \mathbb{R}_+} \phi_{x,i}(t, 0, \mathbf{u}_i^{\text{impulse}}, \overline{\mathbf{d}}_i) \right) \leq \rho_G \tag{B.29}$$

Basically, under control in acceleration with uncertainty, $\sup_{t\in\mathbb{R}_+} \phi_{x,i}(t, 0, \mathbf{u}_i^{\text{impulse}}, \overline{\mathbf{d}}_i)$ represents the control resolution as it is the maximum distance traveled by a robot starting from stop and applying an impulse control under maximal disturbance considering all robots. However, compared to Equation (B.9), a new term appears: σ_i^x, the precision on the position. Hence, Equation (B.29) states that the sum of the control resolution and the observation precision on the position needs to be lower than the safety margin. It is not surprising that the precision on the position appears in the liveness condition. Consider the scenario of Figure B.3 where priorities are cyclic. As the uncertainty in position is large, it is very similar to as if robots were much larger than they are. Hence, one can easily see that a deadlock can possibly occur.

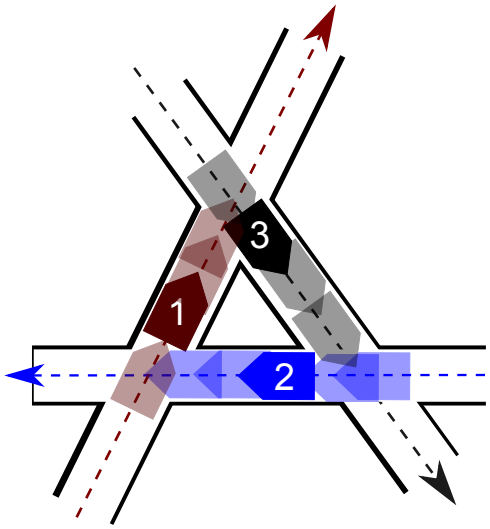


Figure B.3: A three-robot scenario where Condition (B.29) is not respected. Even if the priority graph is feasible, due to a large uncertainty on position, liveness cannot be guaranteed and robots following the control law will be possibly stuck in a deadlock configuration.

Liveness under Condition (B.29) is proved in the sequel.

Theorem B.3.1 (Liveness)**.** *Given a feasible priority graph G with a strictly positive feasibility margin ρ_G satisfying Condition (B.29), an initial brake safe non-deterministic state $\hat{s} \in \hat{B}_G$ and an observation signal $\mathbf{y} \in \mathbf{Y}$, there exists $T > 0$ such that:*

$$\hat{\phi}_x(T, \hat{s}, \hat{g}^G, \mathbf{y}) \in 2^{\chi^{\text{goal}}} \tag{B.30}$$

We start with some lemmas.

Lemma B.3.1. *Given $i \in \mathcal{R}$, $\hat{s}_i \in 2^{S_i}$, $\mathbf{y}_i \in \mathbf{Y}_i$ and $\mathbf{u}_i \in \mathbf{U}_i$ taking values in $\{\underline{u}_i, \overline{u}_i\}$, the following implication holds:*

$$\left[\forall k \in \mathbb{N}, \inf \hat{\phi}_{x,i}(k, \hat{s}_i, \mathbf{u}_i, \mathbf{y}_i) \leq \overline{x}_i^{\text{obs}}\right] \Rightarrow \left[\exists k^0 \in \mathbb{N} : \forall k \geq k^0, \mathbf{u}_i(k) = \underline{u}_i\right] \tag{B.31}$$

Proof. We reason by contraposition. Consider $i \in \mathcal{R}$, $s_i \in S_i$, $\mathbf{y}_i \in \mathbf{Y}_i$ and $\mathbf{u}_i \in \mathbf{U}_i$ taking values in $\{\underline{u}_i, \overline{u}_i\}$. Assume that $\forall k^0 \in \mathbb{N}, \exists k \in \mathbb{N}, k \geq k^0 : \mathbf{u}_i(k) = \overline{u}_i$. Then, we can build an infinite sequence of slots $\{k_1^0, k_2^0, ...\}$ where $\mathbf{u}_i$ takes value $\overline{u}_i$. At each of them, the increase in $\inf \hat{\phi}_{x,i}(k, \hat{s}_i, \mathbf{u}_i, \mathbf{y}_i)$ within the slot is at least of $\phi_{x,i}(1, 0, \overline{\mathbf{u}}_i, \underline{\mathbf{d}}_i) > 0$ (see Assumptions (6.4) and (6.5)). Hence, we have $\lim_{k \to +\infty} \inf \hat{\phi}_{x,i}(k, \hat{s}_i, \mathbf{u}_i, \mathbf{y}_i) = +\infty$ and in particular $\inf \hat{\phi}_{x,i}(k, \hat{s}_i, \mathbf{u}_i, \mathbf{y}_i) > \overline{x}_i^{\text{obs}}$ for some $k \in \mathbb{N}$. □

Lemma B.3.2. *Given $i \in \mathcal{R}$, for all $\hat{s}_i \in 2^{S_i}$ and $\mathbf{y}_i \in \mathbf{Y}_i$, there exists $k^1 \in \mathbb{N}$ such that:*

$$\forall k \geq k^1, \sup \hat{\phi}_{v,i}(k, \hat{s}_i, \underline{\mathbf{u}}_i, \mathbf{y}_i) = 0 \tag{B.32}$$

Proof. Take a priority graph $G \in \mathcal{G}$ satisfying Condition (B.29), a robot $i \in \mathcal{R}$, a non-deterministic state $\hat{s}_i \in 2^{S_i}$ and a control $\mathbf{y}_i \in \mathbf{Y}_i$. Consider the trajectory $t \mapsto \phi(t, \hat{s}_i, \underline{\mathbf{u}}_i, \mathbf{y}_i)$ of robot i under control $\underline{\mathbf{u}}_i$. At each time slot, there are two options:

- $\sup \hat{\phi}_{v,i}(t, \hat{s}_i, \underline{\mathbf{u}}_i, \mathbf{y}_i)$ vanishes within the slot;
- the variation of $\sup \hat{\phi}_{v,i}(t, \hat{s}_i, \underline{\mathbf{u}}_i, \mathbf{y}_i)$ within the slot is lower than $\underline{u}_i + \overline{d}_i < 0$ (see Assumption (6.6)).

In conclusion, $\sup \hat{\phi}_{v,i}(t, \hat{s}_i, \underline{\mathbf{u}}_i, \mathbf{y}_i)$ vanishes in finite time. □

Lemma B.3.3. *Consider a non-deterministic state $\hat{s} \in 2^S$ included in an open parallelepiped whose side along axis i has length σ_i. Then, for all $i \in \mathcal{R}$ satisfying $\pi_{v,i}(\sup \hat{s}) = 0$, we have:*

$$\hat{g}_i^G(\hat{s}) := \begin{cases} \underline{u}_i & \text{if } \exists (j,i) \in E(G_{|\pi_x(\inf \hat{s}), \rho_G}), \exists t \geq 0 \text{ s.t. } \phi_x(t, \widetilde{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \in \chi_{j \succ i}^{\text{obs}} \\ \overline{u}_i & \text{else.} \end{cases} \tag{B.33}$$

Proof of Lemma B.3.3. Take $\hat{s} \in 2^S$ and $i \in \mathcal{R}$ satisfying $\pi_{v,i}(\sup \hat{s}) = 0$, take $i \in \mathcal{R}$ and $(j,i) \in E(G)$. By construction of $\hat{g}^G$, we have:

$$\hat{g}_i^G(\hat{s}) := \begin{cases} \underline{u}_i & \text{if } \exists (j,i) \in E(G), \exists t \geq 0 \text{ s.t. } \phi_x(t, \widehat{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \in \chi^{\text{obs}}_{j \succ i} \\ \overline{u}_i & \text{else.} \end{cases} \tag{B.34}$$

Hence, we have to prove that for all $(j,i) \in E(G)$, if $\phi_x(t, \widehat{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \in \chi^{\text{obs}}_{j \succ i}$ for some $t \geq 0$, then we have $(j,i) \in E(G_{|\pi_x(\inf \hat{s}), \rho_G})$. We will prove this by contraposition.

Assume that $(j,i) \notin E(G_{|\pi_x(\inf \hat{s}), \rho_G})$, then we have:

$$\pi_x(\inf \hat{s}) + \rho_G \mathbf{e}_i \in \chi^{\text{free}}_{j \succ i} \tag{B.35}$$

By assumption, we have $\pi_{v,i}(\sup \hat{s}) = 0$, hence $\pi_{v,i}(\widehat{\sup}^i_i(\hat{s})) = 0$. As the length of $\hat{s}$ along axis x_i is σ^x_i, we have $\pi_{x,i}(\widehat{\sup}^i_i(\hat{s})) \leq \pi_x(\inf \hat{s}_i) + \sigma^x_i$. Hence, by order-preservation, we obtain for all $t \geq 0$:

$$\phi_{x,i}(t, \widehat{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \leq \pi_x(\inf \hat{s}_i) + \sigma^x_i + \phi_{x,i}(t, 0, \tilde{\mathbf{u}}^i_i, \tilde{\mathbf{d}}^i_i) \tag{B.36}$$

By Equation (B.29), we obtain:

$$\phi_{x,i}(t, \widehat{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \leq \pi_x(\inf \hat{s}_i) + \rho_G \tag{B.37}$$

As $t \mapsto \phi_j(t, \widehat{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i)$ is non-decreasing, we obtain:

$$\phi_{x,j}(t, \widehat{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \geq \pi_x(\inf \hat{s}_j) \tag{B.38}$$

By Property 2.2.1, Equation (B.35) implies that $\phi_x(t, \widehat{\sup}^i(\hat{s}), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \in \chi^{\text{free}}_{j \succ i}$ for all $t \geq 0$. □

Lemma B.3.4. *Given $\hat{s} \in 2^S$, for all $i, j \in \mathcal{R}$, for all $\mathbf{u} \in \mathbf{U}$ and $\mathbf{d} \in \mathbf{D}$, the following implication holds:*

$$\left[\exists t \geq 0 : \phi_x(t, \widehat{\sup}^i(\hat{s}), \mathbf{u}, \mathbf{d}) \in \chi^{\text{obs}}_{j \succ i}\right] \Rightarrow \left[\pi_x(\inf \hat{s}_j) < \overline{x}^{\text{obs}}_j\right] \tag{B.39}$$

Proof. Take $\hat{s} \in 2^S$, $i, j \in \mathcal{R}$, $\mathbf{d} \in \mathbf{D}$ and $\mathbf{u} \in \mathbf{U}$. The flow $t \mapsto \phi_x(t, \widehat{\sup}^i(\hat{s}), \mathbf{u}, \mathbf{d}))$ is non-decreasing. Hence, we have:

$$\phi_{x,j}(t, \widehat{\sup}^i(\hat{s}), \mathbf{u}, \mathbf{d}) \geq \phi_{x,j}(0, \widehat{\sup}^i(\hat{s}), \mathbf{u}, \mathbf{d}) = \pi_x(\widehat{\sup}^i_j(\hat{s})) = \pi_x(\inf \hat{s}_j) \tag{B.40}$$

Assume that $\phi_x(t, \widehat{\sup}^i(\hat{s}), \mathbf{u}, \mathbf{d}) \in \chi^{\text{obs}}_{j \succ i}$ for some $t \geq 0$. It implies that $\phi_{x,j}(t, \widehat{\sup}^i(\hat{s}), \mathbf{u}, \mathbf{d}) < \sup\{x_j : x \in \chi^{\text{obs}}_{j \succ i}\} \leq \overline{x}^{\text{obs}}_j$. By transitivity, we obtain $\pi_x(\inf \hat{s}_j) < \overline{x}^{\text{obs}}_j$. □

Proof of Theorem B.3.1. Take a priority graph $G \in \mathcal{G}$ satisfying Condition (B.29). Consider the set $I(t)$ defined as follows:

$$\forall t \in \mathbb{R}_+, I(t) := \left\{ i \in \mathcal{R} : \inf \hat{\phi}_{x,i}(t, \hat{s}, \hat{g}^G, \mathbf{y}) < \overline{x}_i^{\text{obs}} \right\} \tag{B.41}$$

We prove Theorem B.3.1 by contradiction. Theorem B.3.1 is equivalent to the existence of $T > 0$ such that:

$$I(T) = \emptyset \tag{B.42}$$

Assume that for all $t \geq 0$, $I(t) \neq \emptyset$. Then, consider $\underline{I}$ defined as follows:

$$\underline{I} := \cap_{t \geq 0} I(t) \neq \emptyset \tag{B.43}$$

We have for all $i \in \underline{I}$,

$$\forall k \in \mathbb{N}, \inf \hat{\phi}_{x,i}(k, \hat{s}, \hat{g}^G, \mathbf{y}) < \overline{x}_i^{\text{obs}} \tag{B.44}$$

By Lemma B.3.1, a necessary condition for the above equation to be satisfied is that for all $i \in \underline{I}$, there exists $k_i^0 \in \mathbb{N}$ such that:

$$\forall k \geq k_i^0, \hat{g}_i^G(\hat{\phi}(k, \hat{s}, \hat{g}^G, \mathbf{y})) = \underline{u}_i \tag{B.45}$$

Hence, for $k^0 := \max_{i \in \underline{I}} k_i^0$, we have:

$$\forall i \in \underline{I}, \forall k \geq k^0, \hat{g}_i^G(\hat{\phi}(k, \hat{s}, \hat{g}^G, \mathbf{y})) = \underline{u}_i \tag{B.46}$$

By Lemma B.3.2, the velocity of all robots $i \in \underline{I}$ will vanish in finite time. There exists $k^1 \geq k^0$ such that:

$$\forall k \geq k^1, \hat{g}_i^G(\hat{\phi}(k, \hat{s}, \hat{g}^G, \mathbf{y})) = \underline{u}_i \tag{B.47}$$

$$\sup \hat{\phi}_{v,i}(k, \hat{s}, \hat{g}^G, \mathbf{y}) = 0 \tag{B.48}$$

Take $k \geq k^1$ big enough, so that $I(k) = \underline{I}$. By Lemma B.3.3, for all $i \in \underline{I}$, there exists $(j, i) \in E(G_{|\inf \hat{\phi}_x(k, \hat{s}, \hat{g}^G, \mathbf{y}), \rho_G})$ such that $\phi_x(\mathbb{R}_+, \widetilde{\sup}^i(\hat{\phi}(k, \hat{s}, \hat{g}^G, \mathbf{y})), \tilde{\mathbf{u}}^i, \tilde{\mathbf{d}}^i) \cap \chi_{j \succ i}^{\text{obs}} \neq \emptyset$ and we have necessarily $\inf \hat{\phi}_{x,j}(k, \hat{s}, \hat{g}^G, \mathbf{y}) < \overline{x}_j^{\text{obs}}$ by Lemma B.3.4, i.e., $j \in I(k) = \underline{I}$. As a result, $G_{|\inf \hat{\phi}_x(k, \hat{s}, \hat{g}^G, \mathbf{y}), \rho_G}$ should be cyclic (there would be a cycle involving vertices in $\underline{I}$). It is absurd. □

Résumés en français

Coordination de robots mobiles par affectation de priorités

RESUME : Depuis la fin des années 1980, le développement de véhicules autonomes est un champ de recherche très actif dans la plupart des grands pays industrialisés. La diminution du nombre d'accidents, des temps de trajet plus courts, une meilleure efficacité énergétique et des besoins en infrastructure plus limités, sont autant d'effets socio-économiques espérés de leur déploiement. Des formes de coopération inter-véhicules et entre les véhicules et l'infrastructure routière sont nécessaires au fonctionnement sûr et efficace du système de transport dans sa globalité. Cette thèse s'intéresse à une forme de coopération particulière en étudiant la coordination de robots mobiles aux intersections. La majorité des systèmes de coordination existants planifie une trajectoire que les robots doivent exécuter afin d'assurer l'absence de collision. C'est une approche classique de la planification, qui est alors considérée comme un mécanisme de génération de l'action. Dans cette thèse, seules les priorités entre les robots sont planifiées – l'ordre relatif de passage dans l'intersection – ce qui est bien plus faible car un grand nombre de trajectoires respectent les même priorités. Plus précisément, les priorités encodent les classes d'homotopie de solutions au problème de coordination. Affecter les priorités revient à choisir une certaine classe d'homotopie pour résoudre le problème de coordination sans planifier une trajectoire particulière. Une fois les priorités affectées, les robots suivent une loi de contrôle qui s'assure de leur respect, c'est-à-dire que la trajectoire effectuée appartient bien à la classe d'homotopie choisie. Il en découle un système de coordination robuste, capable de gérer toute une classe d'événements imprévisibles de façon réactive, ce qui est particulièrement adapté à la coordination de véhicules autonomes aux intersections où voitures, transports en commun et piétons partagent la route.

PARTIE I : La première partie de cette thèse s'intéresse à définir formellement une notion de priorité pour le problème de coordination de robots mobiles le long de chemins fixés. Dans le langage commun des automobilistes, la priorité entre deux véhicules définit leur ordre de passe. Pour définir formellement le concept dans le problème de coordination qui nous intéresse, nous utilisons l'approche standard par l'espace de coordination qui transforme le problème de coordination dans l'espace réel en un problème de recherche de chemin dans un espace abstrait appelé l'espace de coordination, de dimension n où n désigne le nombre de véhicule coordonnés. L'évitement de collision dans l'espace réel est équivalent à l'évitement d'une zone de collision dans l'espace de coordination. Cette dernière a une structure bien particulière, puisqu'elle est constituée d'une union de cylindres de collision, chacun correspondant aux collisions potentielles entre chaque couple de robots. La contribution de cette première partie consiste à utiliser cette structure géométrique particulière afin d'équiper l'espace de coordination d'une notion de priorité. Nous proposons un nouveau cadre théorique – basé sur la notion de priorité – pour étudier le problème de coordination. Comme un chemin dans l'espace de coordination doit éviter chaque cylindre de collision, il doit nécessairement passer d'un côté ou de l'autre de ce dernier. Ainsi, nous montrons qu'un chemin sans collision dans l'espace de coordination induit une relation binaire entre chaque couple de robots. Il apparait alors que choisir de quel coté passer par rapport à un cylindre de collision est équivalent à choisir l'ordre de passage entre deux robots, justifiant ainsi d'appeler priorité cette relation binaire. Nous montrons alors deux résultats fondamentaux de la relation de priorité. Le premier stipule que l'ensemble des chemins sans collision dans l'espace de configuration respectant des priorités

données forme une classe de chemins homotopiques, c'est-à-dire continument déformables les uns vers les autres tout en respectant les priorités données. Ainsi, fixer les priorités revient à choisir une classe d'homotopie particulière pour résoudre le problème de coordination : les priorités structurent topologiquement l'espace de coordination. Le second résultat porte sur la question suivante: existe-t-il nécessairement une solution au problème de coordination respectant des priorités données ? Nous mettons en évidence que des cycles dans les priorités peuvent générer des chaînes de blocage où chaque robot de la chaîne est arrêté et attend le suivant pour pouvoir continuer la traversée de l'intersection. Cependant, l'absence de cycles n'est pas une condition nécessaire à l'existence d'une solution respectant des priorités données, et nous présentons une condition nécessaire et suffisante – une caractérisation – des priorités dites faisables, c'est-à-dire pour lesquelles il existe bien une solution au problème de coordination satisfaisant les priorités données. Cette caractérisation raffine le rôle des cycles dans la formation de blocage. Une façon géométrique de comprendre pourquoi tous les graphes de priorités ne sont pas faisables est de noter que selon la position relative des cylindres de collision dans l'espace de configuration, le choix de passer d'un coté ou de l'autre de chaque cylindre, ne peut pas toujours se faire indépendamment pour chacun d'eux. Enfin, nous montrons que choisir des priorités faisables résout entièrement le problème de l'évitement de blocage : plus précisément, soit les priorités choisies ne sont pas faisables et le système va inévitablement atteindre une configuration de blocage, soit elles sont faisables auquel cas il est garanti que le système ne peut jamais se trouver bloqué. C'est une propriété très intéressante dans la mesure où l'évitement de blocage est un problème complexe, qu'il est difficile de résoudre de façon réactive. En définitive, cette partie de la thèse justifie l'utilisation des priorités comme plan de coordination. Contrairement aux approches traditionnelles de planification de trajectoires, le plan est ici le graphe de priorités résumant de façon synthétique la stratégie de coordination en précisant l'ordre relatif de passage entre les robots, et non pas leurs trajectoires spatio-temporelles. Ce plan ne peut être simplement exécuté par un contrôleur en charge du suivi de trajectoires puisque les trajectoires proprement dites ne sont pas planifiées. Le plan de coordination doit être interprété, utilisé par les robots comme une ressource de coordination afin de traverser l'intersection. Ceci est le sujet de la seconde partie de la thèse qui s'intéresse au contrôle de robots à priorités fixées.

PARTIE II : Dans la seconde partie de cette thèse, nous supposons disposer d'un système de robots mobiles se déplaçant le long de chemins fixés et dont les priorités telles que définies en première partie sont fixées et supposées faisables. Nous nous proposons alors de mettre au point une loi de contrôle permettant aux robots de traverser l'intersection en respectant les priorités données. La loi de contrôle doit assurer que le chemin parcouru par le système dans l'espace de coordination reste bien dans la classe d'homotopie correspondant aux priorités fixées. Dans un premier temps, les robots sont supposés contrôlés en vitesse et une loi de type Bang-Bang est proposée. Celle-ci consiste pour chaque robot à tenter d'avancer à vitesse maximale et à s'arrêter aussi tard que possible, juste à temps pour laisser passer un robot prioritaire. Cette loi de contrôle garantit le respect des priorités, l'optimalité à priorités fixées de la trajectoire résultante, et assure également que tous les robots traverseront l'intersection en temps fini pourvu que les priorités fixées soient bien faisables. Dans ces premiers résultats, l'inertie des robots est totalement négligée puisqu'ils sont supposés être capables de s'arrêter instantanément. C'est pourquoi une loi de contrôle pour des robots contrôlés en accélération est ensuite proposée, intégrant les distances de freinage, et consistant à anticiper le besoin de freiner pour laisser passer des robots prioritaires. La sécurité et la traversée en temps fini sont à nouveau garanties. D'autre part, cette loi de contrôle présente une

robustesse particulière vis-à-vis de toute une classe d'événements imprévus qui nécessitent le freinage d'un, de plusieurs, ou de tous les robots. A tout moment, un robot peut freiner pour réagir à un événement imprévu (perte de signal des capteurs, obstacle détecté sur la voie, etc.), et les autres robots sont assurés de respecter malgré tout les priorités fixées en appliquant simplement la loi de contrôle. C'est une propriété originale en comparaison avec les méthodes standards de coordination par planification de trajectoires qui nécessitent soit le suivi de la trajectoire planifiée – ce qui n'est pas possible si un robot doit freiner pour réagir à un événement imprévu –, soit une replanification coûteuse en temps de calcul et ne garantissant pas une réponse du système en temps réel. Le dernier chapitre de cette partie s'intéresse enfin aux cas de robots toujours contrôlés en accélération, mais où des incertitudes bornées sont présentes à la fois sur le contrôle des robots et sur les observations. Dans ce cadre non déterministe, la loi de contrôle proposée présente toujours les propriétés recherchées de sécurité, de robustesse et de garantie de traversée en temps fini.

PARTIE III : La première partie de cette thèse a suggéré l'utilisation des priorités pour guider les robots au sein de l'intersection et la deuxième partie a fourni des solutions au contrôle de robots à priorités fixées, laissant de coté de nombreux aspects de la mise au point d'un système de coordination multi-robot. Tout d'abord, un tel système est ouvert puisque des robots arrivent dans l'intersection et en sortent au cours du temps. Ainsi, les priorités doivent être affectées de façon dynamique. De plus, l'affectation des priorités et le contrôle des robots à priorités fixées doivent s'exécuter en parallèle. La dernière partie de cette thèse se propose donc de fournir une architecture de contrôle pour un système de coordination de robots mobiles basé sur l'affectation de priorités. Nous adoptons une architecture hybride comprenant trois couches : une couche purement délibérative gérée par une infrastructure centrale, une couche purement réactive distribuée sur chaque robot, et une couche intermédiaire – le séquenceur – également distribuée et jouant un rôle d'interface. Le contrôleur d'intersection est l'infrastructure centrale en charge d'autoriser les robots à pénétrer dans la zone de contrôle et d'affecter les priorités. C'est une tâche purement délibérative puisqu'elle consiste à raisonner sur le futur pour effectuer un choix efficace d'affectation des priorités. La tâche du séquenceur des robots est de communiquer avec le contrôleur de l'intersection afin d'obtenir le droit d'entrer dans la zone de contrôle et d'être affecté une priorité vis-à-vis des autres robots. La couche réactive implémentée par les robots consiste en un ensemble de comportements simples implémentés par chacun d'eux. Tout d'abord, un robot ne doit pas entrer dans la zone de contrôle tant qu'il n'y a pas été accepté par le contrôleur d'intersection. D'autre part, une fois accepté et affecté une priorité par rapport aux autres robots, il doit implémenter une loi de contrôle assurant le respect des priorités fixées telle que celles présentées en deuxième partie. Enfin, le robot peut également implémenter d'autres comportements lui permettant de réagir à son environnement comme la détection d'obstacles ou de piétons sur la voie, l'amenant par exemple à freiner. La robustesse au freinage de la loi de contrôle présentée en deuxième partie apparait alors particulièrement intéressante puisqu'elle autorise un comportement du robot à requérir un freinage – par exemple pour éviter un obstacle sur la voie – tout en conservant la garantie du respect des priorités, et donc l'absence de collisions. De plus, une fois que l'élément requérant le freinage a disparu, le système peut redémarrer en conservant les mêmes priorités et en appliquant la même loi de contrôle sans replanification. Les simulations présentées dans cette dernière partie tentent de montrer les avantages qualitatifs significatifs, en termes de robustesse, du système de coordination proposé par rapport à l'état de l'art.

Coordination de robots mobiles par affectation de priorités

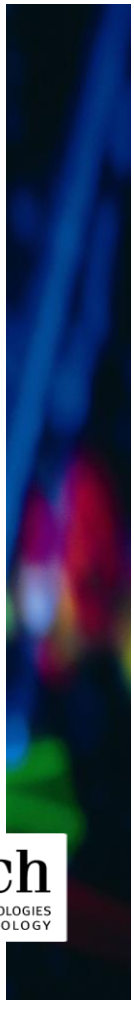
RESUME: Depuis la fin des années 1980, le développement de véhicules autonomes est un champ de recherche très actif dans la plupart des grands pays industrialisés. La diminution du nombre d'accidents, des temps de trajet plus courts, une meilleure efficacité énergétique et des besoins en infrastructure plus limités, sont autant d'effets socio-économiques espérés de leur déploiement. Des formes de coopération inter-véhicules et entre les véhicules et l'infrastructure routière sont nécessaires au fonctionnement sûr et efficace du système de transport dans sa globalité. Cette thèse s'intéresse à une forme de coopération particulière en étudiant la coordination de robots mobiles aux intersections. La majorité des systèmes de coordination existants planifie une trajectoire que les robots doivent exécuter afin d'assurer l'absence de collision. C'est une approche classique de la planification, qui est alors considérée comme un mécanisme de génération de l'action. Dans cette thèse, seules les priorités entre les robots sont planifiées – l'ordre relatif de passage dans l'intersection – ce qui est bien plus faible car un grand nombre de trajectoires respectent les même priorités. Plus précisément, les priorités encodent les classes d'homotopie de solutions au problème de coordination. Affecter les priorités revient à choisir une certaine classe d'homotopie pour résoudre le problème de coordination sans planifier une trajectoire particulière. Une fois les priorités affectées, les robots suivent une loi de contrôle qui s'assure de leur respect, c'est-à-dire que la trajectoire effectuée appartient bien à la classe d'homotopie choisie. Il en découle un système de coordination robuste, capable de gérer toute une classe d'événements imprévisibles de façon réactive, ce qui est particulièrement adapté à la coordination de véhicules autonomes aux intersections où voitures, transports en commun et piétons partagent la route.

Mots clés : robots mobiles, systèmes multi-robot, planification de mouvement, espace de coordination, graphe de priorités, classe d'homotopie, sécurité, robustesse, architecture hybride

Priority-Based coordination of mobile robots

ABSTRACT: Since the end of the 1980's, the development of self-driven autonomous vehicles is an intensive research area in most major industrial countries. Positive socio-economic potential impacts include a decrease of crashes, a reduction of travel times, energy efficiency improvements, and a reduced need of costly physical infrastructure. Some form of vehicle-to-vehicle and/or vehicle-to-infrastructure cooperation is required to ensure a safe and efficient global transportation system. This thesis deals with a particular form of cooperation by studying the problem of coordinating multiple mobile robots at an intersection area. Most of coordination systems proposed in previous work consist in planning a trajectory and to control the robots along the planned trajectory: that is the plan-as-program paradigm where planning is considered as a generative mechanism of action. The approach of the thesis is to plan priorities – the relative order of robots to go through the intersection – which is much weaker as many trajectories respect the same priorities. More precisely, priorities encode the homotopy classes of solutions to the coordination problem. Priority assignment is equivalent to the choice of some homotopy class to solve the coordination problem instead of a particular trajectory. Once priorities are assigned, robots are controlled through a control law preserving the assigned priorities, i.e., ensuring the described trajectory belongs to the chosen homotopy class. It results in a more robust coordination system – able to handle a large class of unexpected events in a reactive manner – particularly well adapted for an application to the coordination of autonomous vehicles at intersections where cars, public transport and pedestrians share the road.

Keywords : mobile robots, multi robot systems, motion planning, coordination space, priority graph, homotopy class, safety, robustness, hybrid architecture



MINES ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY